



industriales
etsii

**Escuela Técnica
Superior
de Ingeniería
Industrial**

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Escuela Técnica Superior de Ingeniería Industrial

**Modelos en red para la resolución de sistemas
de ecuaciones diferenciales acopladas.
Aplicación a problemas mecánicos.**

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA MECÁNICA



**Universidad
Politécnica
de Cartagena**

Autor: Mario García Pérez
Director: Juan Francisco Sánchez Pérez
Codirector: Francisco Alhama López

Cartagena, 22 de octubre de 2014

ÍNDICE

Objetivos. Capítulo 1	1
1.1 Introducción.....	1
1.2 Objetivos	2
El método de simulación por redes. Capítulo 2.....	4
2.1 El método de simulación por redes (mesir) como método numérico	6
2.2 Simulación. El programa PSpice	7
2.3 La programación con MATLAB	14
Modelos en red. Capítulo 3	17
3.1 Componentes eléctricos y ecuaciones Constitutivas	20
3.1.1 Fuentes independientes	20
3.1.2 Elementos pasivos	20
3.1.3 Fuentes controladas	22
3.2 Diseño de modelos básicos	24
3.2.1 Circuitos asociados a las sucesivas derivadas.....	24
3.2.2 Circuitos asociados al tiempo y a los términos independientes	31
3.2.3 Circuito principal.....	32
3.2.4 Modelos duales mediante bobinas.....	41
El programa CODENS_14. Capítulo 4.....	43
4.1 Estructura de las ecuaciones a resolver	45
4.1.1 ecuaciones diferenciales ordinarias.....	45
4.1.2 Sistemas acoplados de ecuaciones diferenciales ordinarias	47
4.2 El programa CODENS_14.....	48
4.2.1 Interfaz de CODENS	48
4.2.2 Nomenclatura de componentes y nodos.....	66
4.3 Estructura de los archivos de texto	77
4.4 Bloque de programación (GUI) y archivos (m-file) contenidos en CODENS_14	80
Aplicación de ejemplos. Capítulo 5	82
5.1 Problema de masas alineadas con resortes	82
5.2 Problema biológico de tres especies.....	87
Conclusiones y referencias. Capítulo 6.....	93

OBJETIVOS

Capítulo 1

1.1 Introducción

El presente proyecto pertenece a la serie de trabajos desarrollados por el grupo de simulación por redes de la UPCT. Son trabajos relacionados con la simulación numérica de diferentes problemas de ciencia e ingeniería mediante el llamado Método de Simulación por redes. Entre otros campos de aplicación, se han abordado problemas no lineales en transmisión de calor, problemas inversos aplicados a la determinación de propiedades térmicas (conductividad y calor específico) de sólidos y fluidos, incluyendo problemas de cambio de fase o frontera móvil, problemas de flujo y transporte de calor en medios porosos y no porosos, ídem con transporte de sal en aguas subterráneas, asentamiento de terrenos, problemas de tribología, problemas elastostáticos con formulación clásica y formulaciones en potencial y problemas de ingeniería química (diferentes tipos de oxidación).

Los pasos que se siguen para la aplicación del método de redes son los siguientes:

1. Discretización espacial de las ecuaciones de gobierno, diferenciales ordinarias o en derivadas parciales, y condiciones de contorno, reteniendo el tiempo como una variable continua,
2. Elaboración de un modelo en red (o circuito eléctrico) cuyas ecuaciones sean formalmente equivalentes a las ecuaciones diferenciales en diferencias finitas procedentes de la discretización del paso anterior,
3. Simulación del modelo en red en un código o programa apropiado de simulación de circuitos (por ejemplo, Pspice u Orcad),
4. Representación de las soluciones en forma tabulada o gráfica.

Es una herramienta de cálculo que ha demostrado ser muy eficaz en todos los casos, particularmente en escenarios fuertemente no lineales para lo que la solución por métodos más clásicos (elementos finitos y diferencias finitas) no es trivial.

Además, el grupo de investigación ha desarrollado, para alguno de los escenarios anteriores, paquetes de programación o códigos que permiten a cualquier usuario abordar este tipo de problemas con unos conocimientos relativamente elementales. Estos códigos se han desarrollado en lenguajes de programación amenos, en un entorno de ventanas, con código

prácticamente abierto, en donde el usuario tiene acceso al archivo del modelo, para poder manipularlo y ejecutarlo, así como a los entornos propios del programa de cálculo numérico y de las rutinas de programación para el tratamiento de datos finales y su representación gráfica. Los códigos integran, por tanto, todas las rutinas de programación necesarias para la creación de los archivos de texto del modelo y para el tratamiento de datos.

1.2 Objetivos

La presente memoria aplica el método de simulación por redes a la búsqueda de soluciones numéricas de problemas formulados mediante ecuaciones diferenciales ordinarias o sistemas acoplados de estas ecuaciones, mediante el diseño de modelos en red y la subsiguiente simulación de estos modelos en un código de resolución de circuitos tal como Pspice u Orcad.

La elaboración de este trabajo conlleva como objetivo inmediato la familiarización del alumno con conceptos no triviales de la teoría de circuitos eléctricos (aplicación de leyes, teoremas y ecuaciones constitutivas de componentes eléctricos) que le permitan el diseño de éstos a partir de ecuaciones diferenciales, resolviendo las no linealidades y acoplamientos entre ecuaciones.

Además, el alumno deberá estar familiarizado con el lenguaje de programación en entorno de ventanas de Matlab, conocer este programa al nivel de manipulación y representación gráfica de datos tabulados procedentes de los archivos de salida de Pspice. También, debe conocer el funcionamiento del simulador de circuitos Pspice (reglas de programación de archivos de modelos, opciones de simulación y sentencias propias, entorno gráfico propio de salida, estructura de los datos tabulados en los archivos de salida, etc.)

El objetivo del proyecto se puede resumir en finalizar CODENS, un programa general para la simulación de los modelos en red que contenga:

- Las rutinas de programación que elaboren los modelos de red en forma de archivo de texto y los ejecuten en el código de simulación de circuitos Pspice u Orcad.

Estas rutinas permitirán:

- o generar, presentar, modificar y guardar el modelo en red en forma de archivo de texto.
- o arrancar el código de simulación y ejecutar los modelos.

- Mejorar la interface de comunicación a través de ventanas que permita al usuario:
 - o introducir las ecuaciones ordenadamente utilizando una terminología simbólica y el uso de parámetros o constantes numéricas y no numéricas.
 - o introducir el resto de los parámetros que completan el modelo en red, incluyendo las condiciones iniciales, las ventanas de tiempo de la simulación, los datos impresos de salida, etc.
- Eliminar los errores que aparecen en el programa CODENS en su utilización, así como ventanas y acciones incompletas.
- Aplicación de problemas mecánicos mediante la simulación por redes a través del programa CODENS.

En particular, para la asignación de nudos a las soluciones de cada variable y a sus derivadas se seguirá una regla sencilla que asocie la nomenclatura del nudo a la variable asignada.

Por último los gráficos de salida se organizarán en grupos para cada variable. En cada grupo se mostrará simultáneamente pero en gráficos distintos $y(t)$, $dy(t)/dt$, $d^2y(t)/dt^2...$ por un lado y el diagrama de fases $y = f[dy(t)/dt]$, por otro.

EL MÉTODO DE SIMULACIÓN POR REDES Y LA PROGRAMACIÓN EN MATLAB

Capítulo 2

El método de simulación por redes es una técnica para el estudio de cualquier proceso físico que pueda definirse mediante un modelo matemático o conjunto completo de ecuaciones, generalmente en derivadas parciales acopladas o no. Partiendo de éstas el procedimiento consiste en dos etapas bien diferenciadas: en primer lugar, en elaborar un “modelo en red” o circuito eléctrico equivalente al proceso, y en segundo lugar, en simular dicho proceso obteniendo la solución del modelo en red mediante un programa adecuado de resolución de circuitos eléctricos.

En los últimos años el método ha sido aplicado con éxito a campos de investigación, tales como transporte de calor en fluidos, sistemas caóticos, vibraciones mecánicas, elasticidad, problemas inversos, etc., incluyendo el campo que nos ocupa de flujo asociado a densidad variable con transporte de soluto, Alhama y col. [2010] y Soto y col. [2007]), así como en el diseño de programas educativos, Alhama y Del Cerro [2010]).

La equivalencia formal entre el modelo en red y el proceso físico reside en que ambos se rigen por las mismas ecuaciones discretizadas en el espacio, es decir por las mismas ecuaciones diferenciales en diferencias finitas, referidas a un elemento de volumen o celda, y las mismas ecuaciones discretizadas para las condiciones de contorno.

¿Cómo se elabora el modelo en red? La técnica consiste en reticular el espacio en elementos de volumen o celdas elementales; al aplicar a estas celdas de tamaño finito las ecuaciones diferenciales, se obtienen un conjunto de ecuaciones diferenciales en diferencias finitas que se constituyen en el punto de partida para la obtención del modelo en red correspondiente; una vez establecida la correspondencia entre variables dependientes del problema y variables eléctricas, tensiones e intensidades, los resultados de la simulación se pueden interpretar en términos del proceso que se

modela. La asociación de celdas, de acuerdo con la geometría del problema, configura el modelo en red correspondiente a todo el medio finito, que es tanto más preciso cuanto mayor sea el número de éstas (Alhama [1999]). Las condiciones de contorno e iniciales se incorporan al modelo de manera simple.

El planteamiento formal que sirve de base para el desarrollo del MESIR es la “teoría de redes” de Peusner [1987], en la que se apoya su “termodinámica de redes”; la teoría de redes a su vez se sustenta en la teoría de circuitos a partir de una generalización de sus variables conjugadas, corriente eléctrica y diferencia de potencial (d.d.p.). Los modelos en red son para Peusner una representación exacta de las características matemáticas de los procesos que describen. El MESIR es, por otro lado, un método de simulación en tanto que incluye la resolución numérica del modelo en red. Así, las variables flujos y fuerzas características del mismo deben satisfacer las leyes de Kirchhoff, y sus relaciones constitutivas determinarán los elementos de circuito correspondientes. Ahora bien, en cada proceso concreto y una vez elegidas las variables conjugadas, la información de qué elementos de circuito intervienen en el modelo en red y cómo se conectan entre sí, se obtiene del modelo matemático y no de consideraciones de tipo físico acerca del papel que juegan estas variables.

En síntesis, en la teoría de redes, la viabilidad de un modelo en red supone:

- 1) La existencia de una red independiente del tiempo,
- 2) La existencia de una magnitud $j_{N-N'}$ llamada flujo, asociada a cada rama que conecta los nudos $N-N'$ y que va de N a N' . $j_{N-N'}$ obedece las leyes de Kirchhoff para corrientes (LCK), y
- 3) La existencia de una magnitud, ϕ , asociada a cada nudo, tal que la diferencia $X_{N-N'} = \phi_N - \phi_{N'}$, llamada fuerza, obedece la ley de los voltajes de Kirchhoff (LVK).

Las relaciones entre flujo y fuerza asociados a una rama y sus (dos) nudos límite, que pueden incluir o no variaciones temporales de estas variables que se dicen conjugadas, definen los elementos concretos del circuito equivalente a esa rama. La relación causa-

efecto entre las variables conjugadas es completamente arbitraria con tal que sea consistente con 2) y 3).

2.1 El método de simulación por redes (mesir) como método numérico

En el MESIR, el punto de inicio es siempre el modelo matemático de un cierto proceso, esto es, un conjunto de ecuaciones en derivadas parciales (EDP) espacio-temporales; la discretización de la variable espacial permite establecer el modelo en red o red eléctrica equivalente. Esta es la única manipulación directa que se hace de las ecuaciones.

El modelo en red es el formato que se da al modelo matemático para que pueda ser utilizado como entrada (fichero) en un programa de resolución de circuitos eléctricos tal como Pspice®, Nagel [1975 y 1977], Pspice [1994], Vladimirescu [1994] y Kielkowsky [1994]. Este software es el que resuelve las ecuaciones de la red proporcionando la solución numérica del modelo matemático.

En definitiva, puesto que la simulación del modelo en red mediante ordenador implica la resolución numérica de sus ecuaciones, el método de simulación por redes es, de hecho, un método numérico. A continuación exponemos las diferencias de estrategias más notables al compararlo con otros métodos numéricos.

Cuando en una ecuación en derivadas parciales se hace una doble reticulación, espacial y temporal, se reemplazan de hecho las derivadas parciales por aproximaciones algebraicas, lo que conduce a un conjunto de 'ecuaciones algebraicas' que aproximan las EDP; para la solución numérica de éstas se utiliza un software adecuado de matemáticas. Este procedimiento es la base de los bien conocidos métodos numéricos de diferencias finitas, elementos finitos y volúmenes finitos, para la solución de las EDP.

Como ya se ha comentado, la elaboración del modelo en red pasa por la reticulación espacial, pero no temporal. Se parte, pues, de un sistema de ecuaciones en derivadas

parciales cuya reticulación espacial las convierte en ecuaciones diferenciales ordinarias en el tiempo, que son las del circuito correspondiente a una celda elemental. La diferencia esencial es, pues, que en los métodos numéricos convencionales se realiza una reticulación simultánea de las dos variables independientes, espacio y tiempo, mientras que en el MESIR la reticulación es sucesiva; 1ª etapa, una reticulación espacial de la que se obtiene el modelo en red y 2ª etapa, una reticulación temporal, realizada por el propio software en el proceso de simulación.

En el MESIR, previa la definición de la variable flujo, $j(q,t)=\partial\phi(q,t)/\partial q$, las EDP toman la forma

$$f_i [\phi, \partial\phi/\partial t, \partial^2\phi/\partial t^2, j, \partial j/\partial q, \partial j/\partial t, q, t] = 0$$

que con la discretización espacial se convierten en

$$F_i [\phi, d\phi/dt, d^2\phi/dt^2, j, dj/dt, t] = 0$$

que son las ecuaciones del circuito (la conexión entre $j(q,t)$ y $\phi(q,t)$ no se deshace).

Si $j(q,t)=\partial\phi(q,t)/\partial q_i$ no es una definición, sino una relación física entre variables definidas independientemente, la red puede considerarse como una descripción alternativa del sistema. Si además j corresponde a un flujo de transporte de una cierta magnitud, amén de lo anterior, los elementos del circuito y ciertos parámetros derivados del conjunto de la red (como la impedancia) pueden dotarse de un significado físico equivalente al que tienen en el transporte de la carga eléctrica. En estos casos es evidente que el MESIR proporciona más información que la estricta respuesta numérica del sistema.

2.2 Simulación. El programa Pspice

Una vez obtenido el modelo en red se procede a su análisis. Para ello hemos buscado un software adecuado para la solución de circuitos eléctricos Pspice [1994]. Las dificultades estarán, inicialmente, en el aprendizaje de un lenguaje de programación, que puede ser de texto o gráfico; sin embargo se trata de dificultades fáciles de salvar dado que son muy pocas las reglas de programación que se requieren (téngase en

cuenta que los modelos están constituidos por muy pocos dispositivos como se verá en el siguiente capítulo).

Pspice ha sido utilizado por otros autores para resolver problemas de otras disciplinas. Baker y Shortt [1990] estudian el comportamiento de componentes integrados en diferentes rangos de temperatura, Bello [1991] lo aplica a la resolución de problemas mecánicos, Herber [1992] lo aplica a la resolución de ecuaciones diferenciales ordinarias, Hamill [1993], a problemas estadísticos y relacionados con el caos, etc.

En el proceso de simulación el circuito se presenta al ordenador como un conjunto de ecuaciones matemáticas y éste, mediante procedimientos de análisis numérico, proporciona toda la información solicitada por el investigador para cada tipo de análisis. De esta forma se obtienen los datos correspondientes a medidas típicas de laboratorio con un margen de error despreciable y sin afectar al circuito; más aún, pueden alterarse las condiciones iniciales, de contorno, y las características térmicas del medio con sencillos cambios en el programa, y el análisis puede aportar datos sobre el comportamiento del circuito más allá de los límites que virtualmente se pueden obtener con medidas reales.

La simulación está estructurada en cinco subprogramas principales, que interaccionan entre ellos a través de una estructura de datos que es almacenada en un área común del programa. Estos subprogramas son: entrada, organización, análisis, salida y utilidades, Figura 2.1.

El subprograma de entrada lee el archivo de entrada, construye una estructura de datos y chequea el circuito. El de organización, una vez que el programa se ha ejecutado con éxito, construye las estructuras adicionales de datos que serán requeridas en el programa de análisis, parte esencial de la simulación. El subprograma de salida genera y organiza, en la memoria central, los resultados solicitados por el usuario en forma tabulada o gráfica. Las utilidades son aspectos secundarios no relacionados directamente con la simulación; éstas permiten, por ejemplo, almacenar componentes o partes de modelos para ser compartidos por otros usuarios.

El subprograma análisis es la parte más importante del programa de simulación. Ejecuta los análisis del circuito requeridos, de acuerdo con las indicaciones del archivo de entrada; la información resultante se almacena en la memoria central para su posterior procesamiento en los archivos de salida. Mientras que la facilidad de uso del programa reside en los subprogramas de entrada y salida, el programa de análisis, que contiene algoritmos más complejos y consume la fracción mayor del tiempo de computación, determina la eficiencia de la simulación.

En el proceso de simulación, se obtiene la solución numérica de la representación matemática del modelo en red. Esta contiene:

- 1) Las ecuaciones matemáticas de los diferentes tipos de monpuertas.
- 2) Las ecuaciones correspondientes a las restricciones impuestas por las leyes de Kirchhoff, propias de la teoría de circuitos, que han de satisfacerse entre las ramas y nudos del circuito.
- 3) La información particular sobre la interconexión de los diferentes componentes eléctricos de cada modelo. Toda esta información compone un extenso sistema de ecuaciones algebraico-diferenciales del tipo.

$$F(\phi, j, d\phi/dt, dj/dt, t) = 0$$

donde $\phi = (\phi_1, \phi_2, \dots, \phi_n)$ y $j = (j_1, j_2, \dots, j_m)$ son vectores formados por las variables dependientes del circuito, corrientes y tensiones, y F , en general, un operador no lineal.

El conjunto de tareas que componen el proceso de simulación puede ser agrupado en los siguientes tópicos (o algoritmos de computación), i) formulación de las ecuaciones, ii) solución de ecuaciones lineales, iii) solución de ecuaciones no lineales y iv) integración numérica.

Pspice es miembro de la familia de programas de simulación de circuitos Pspice2 (Nagel, [1975]). Mucho más potente y rápido que sus predecesores, fue desarrollado en la Universidad de California en los años setenta y utiliza algoritmos numéricos más refinados con formatos de entrada-salida idénticos.

El uso extendido de Pspice da fe de su capacidad para el tratamiento de una extensa variedad de problemas en simulación de circuitos, resolviendo:

- 1) respuesta en corriente continua,
- 2) respuesta transitoria en el tiempo y análisis de Fourier en el dominio de la frecuencia, y
- 3) análisis de pequeña señal en corriente alterna y distorsión.

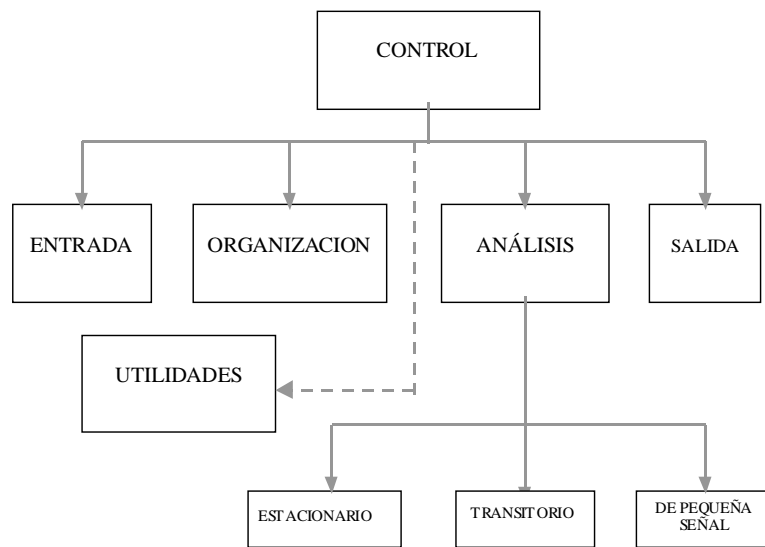


Figura 2.1 Diagrama bloques del programa de simulación de circuitos Pspice

De estas posibilidades nuestro interés se ciñe al estudio de transitorios y estacionarios.

En el análisis de continua Pspice determina el punto de trabajo, es decir, los valores de polarización de sus componentes en ausencia de excitaciones alternas. Para este cálculo se elimina la acción de los condensadores y bobinas, los primeros quedan como circuitos abiertos y las bobinas se cortocircuitan.

Para el análisis transitorio Pspice parte del intervalo de tiempo $(0,T)$ solicitado, que puede ser menor o mayor que la duración del transitorio, y facilita los datos en forma de listado o mediante gráficas. Si los resultados se quieren en forma tabular el usuario debe indicar el instante inicial, el final, el paso temporal y el número de variables listadas; si se solicitan en forma gráfica, una sentencia de programa permite organizarlos y almacenarlos para ser utilizados con ese propósito en cada momento.

La formulación de las ecuaciones del circuito se realiza mediante el método conocido como Análisis Nodal Modificado que permite crear un sistema de ecuaciones casi simétrico que se resuelve con un esfuerzo computacional comparable a otros métodos más simples.

La capacidad de análisis transitorio en un programa de simulación requiere: i) transformar las ecuaciones diferenciales de las monpuertas con capacidad de almacenamiento energético en un sistema de ecuaciones algebraicas correspondientes a un estado de “cuasi corriente-continua”; este proceso proporciona, mediante un algoritmo de integración, los datos de partida de las mencionadas monpuertas cada vez que se produce un incremento de la variable tiempo, ii) fijar el error local de truncamiento (dependiente directamente del intervalo de tiempo) en el algoritmo de integración anterior, y reajustar el intervalo de tiempo mediante iteración hasta conseguir niveles de error prácticamente nulos.

Para la solución del sistema matricial de ecuaciones lineales se utiliza el método de factorización LU que elimina directamente las incógnitas (este método descompone la matriz de coeficientes en producto de matrices triangulares, “lower and upper, LU”, cuyas inversas no precisan ser calculadas, lo que redundaría en un menor esfuerzo computacional). Para minimizar el esfuerzo de cálculo, las ecuaciones se reordenan usando el método de Markowitz.

La implantación del método de Newton-Raphson para resolver las ecuaciones no lineales requiere de algunas modificaciones debido a la naturaleza exponencial de las ecuaciones que modelan ciertos dispositivos eléctricos y que pueden dar lugar a “overflow” numéricos. Nagel, en su tesis, concluye que el método de Colom, encuadrado dentro de los llamados “simple-limitingmethods” es el más fácil de implementar y requiere el menor número de ecuaciones para converger; además, se incorpora un nuevo criterio de convergencia basado en el grado de aproximación de las ecuaciones de rama linealizadas respecto de las ecuaciones originales no linealizadas, criterio que es independiente de qué variables del circuito sean elegidas como incógnitas en el algoritmo de formulación de las ecuaciones.

Los métodos de integración implantados en Pspice incorporan un proceso de variación dinámica del paso del tiempo de integración para mantener una razonable exactitud en la solución y un tiempo mínimo de computación. Pspice utiliza unos métodos de integración polinomiales basados en el análisis de error de truncamiento local y en la estabilidad (propiedades contrapuestas).

Debido a que ciertos circuitos (que contienen constantes de tiempo de valores muy diferentes) pueden dar lugar a un sistema de ecuaciones “stiff”, es conveniente que el algoritmo de integración sea “stiff-estable”. Para conseguir este objetivo se utilizan métodos de integración trapezoidal y Gear de orden variable de dos a seis.

Tras conseguir la convergencia, la solución se almacena y se reinicia el proceso para el instante siguiente. El paso δt es, en consecuencia, variable de unos tramos del intervalo a otros; el programa los ajusta en función de la precisión exigida a los resultados de manera que el tiempo de computación sea el mínimo. Los datos de simulación correspondientes a tiempos fuera del conjunto discreto de instantes $0, t_1, t_2, \dots, T$ se obtienen por interpolación.

La Figura 2.2 representa un diagrama de flujo que ilustra los cuatro algoritmos de computación que tienen lugar en la simulación de un proceso transitorio (para simplificar se ha supuesto un δt constante). Para el análisis de continua el lazo externo de la figura no se utiliza y, si el circuito contiene sólo elementos lineales, una sola iteración proporciona los resultados finales por lo que el lazo interior puede eliminarse.

La sintaxis de entrada no requiere especiales disposiciones ordenadas de datos, su estilo puede catalogarse más bien como libre y dispone de una razonable fuente de datos que se adjudican por omisión a los componentes cuando éstos no se especifican en detalle. También realiza un buen número de chequeos para asegurar que el circuito ha sido introducido correctamente y el resto de las sentencias de programa están bien escritas, advirtiendo al programador de posibles errores mediante mensajes previos a la ejecución. En definitiva, un usuario principiante necesita especificar un número mínimo de parámetros y controles de simulación para extraer unos resultados de simulación aceptables.

El programa, por fin, se estructura como un listado que contiene todos los componentes eléctricos del circuito (existe la posibilidad de organizar el programa mediante subcircuitos), resistencias, condensadores, fuentes, interruptores, etc., que se introducen uno por uno indicando el nombre, valor, nudos de conexión y el resto de valores de los parámetros característicos.

El software Pspice se programa en su forma clásica por sentencias, elaborando archivos de texto, en un lenguaje relativamente simple (alternativamente es posible elaborar archivos por medio de la opción gráfica 'schematics' seleccionando directamente los elementos de circuito y conectándolos eléctricamente entre si en forma de esquema eléctrico). La sintaxis de entrada no requiere especiales disposiciones ordenadas de datos, su estilo puede catalogarse más bien como libre y dispone de una razonable fuente de datos que se adjudican por omisión a los componentes cuando estos no se especifican en detalle. También realiza un buen número de chequeos para asegurar que el circuito ha sido introducido correctamente y el resto de las sentencias de programa están bien escritas, advirtiendo al programador de posibles errores mediante mensajes previos a la ejecución. En definitiva, un usuario principiante necesita especificar un número mínimo de parámetros y controles de simulación para extraer unos resultados de simulación aceptables.

El programa, por fin, se estructura como un listado que contiene todos los componentes eléctricos del circuito (existe la posibilidad de organizar el programa mediante subcircuitos), resistencias, condensadores, fuentes, interruptores, etc, que se introducen uno por uno indicando el nombre, valor, nudos de conexión y otros parámetros característicos.

El programa Pspice (como, en general, cualquier otro software de resolución de circuitos eléctricos) ofrece muchas posibilidades para el estudio de los sistemas térmicos:

- Permite conocer directamente el estacionario del sistema térmico (BIAS POINT), mediante el análisis en continua del circuito, antes referido. La opción '.TRANS' proporciona el transitorio del proceso,

- Con el análisis en alterna se obtiene de forma inmediata el análisis de respuesta en frecuencia del sistema térmico,
- La aplicación *Probe* muestra de forma gráfica los resultados de la simulación con la máxima precisión que da el programa. Esta aplicación permite también la representación de funciones resultado de operaciones entre variables de la simulación. Por ejemplo, las curvas de la admitancia o la impedancia del sistema (cociente entre corriente y tensión o viceversa) pueden ser directamente obtenidas de *Probe*,
- El software admite la parametrización del modelo en red (sentencia PARAM), lo que constituye un modo ventajoso de utilizar la técnica de cambiar de valores los componentes del circuito para obtener soluciones de problemas similares,
- Las sentencias '.PARAM' y '.STEP' combinadas obtienen la variación secuencial de la respuesta del sistema ante la variación de un parámetro, lo que es una herramienta muy útil para problemas sencillos de optimización (una o dos variables),
- La aplicación *Stimulus* permite la confección de fuentes de tensión o corriente de prácticamente cualquier forma, que pueden representar cualquier estímulo del sistema,
- Admite la *ejecución sucesiva* de programas, técnica que permite arrancar indefinidamente Pspice por otro programa y resolver el circuito para una amplia gama de valores de los componentes. En este caso el programa actúa como un 'solver', cuya misión es resolver las ecuaciones diferenciales del sistema, mientras que al otro programa se le confía la resolución de un problema más amplio.

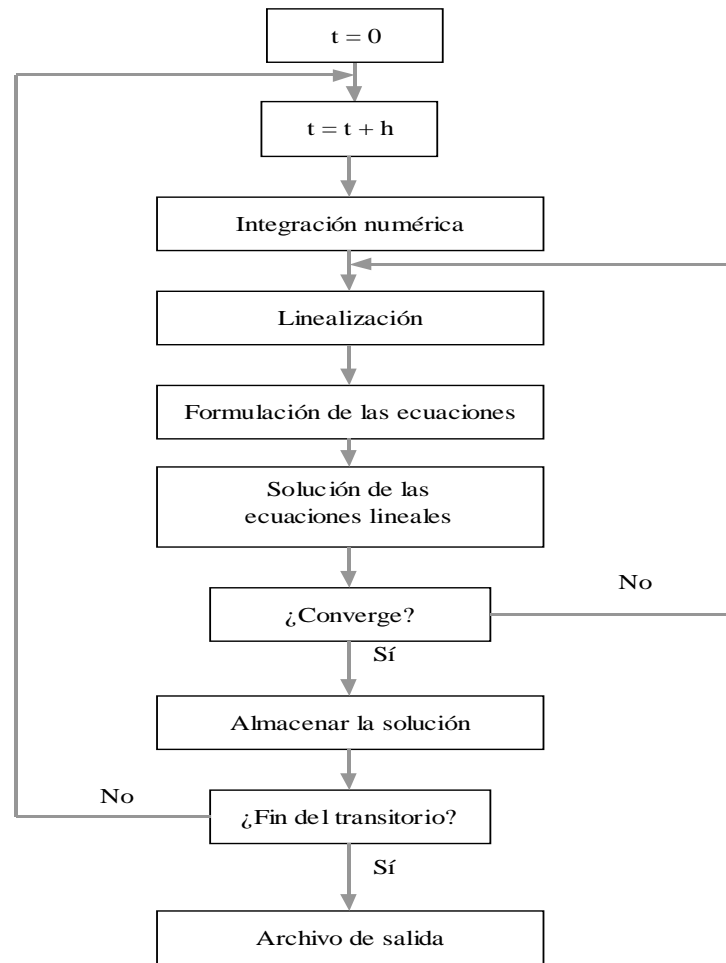
En relación con la presentación de resultados, Pspice permite acceder a los resultados de la simulación (temperaturas y flujos de calor en todo el medio) de dos formas: directamente usando su propio entorno gráfico, muy intuitivo y potente, o accediendo a los archivos de salida de datos los cuales muestran los resultados en forma tabulada; en general estos resultados vienen dados usando como variable independiente el

tiempo, por lo que son muy útiles en problemas transitorios pero no tanto en problemas estacionarios.

2.3 La programación con MATLAB

Se trata de un lenguaje orientado a objetos sencillo, moderno y amigable, con el ambicioso objetivo de recoger las mejores características de muchos otros lenguajes y combinarlas en uno solo en el que se unen la alta productividad y facilidad de aprendizaje. La programación en MATLAB incorpora elementos tales como sistema de tipos homogéneo, propiedades, indexadores, tablas multidimensionales, operadores redefinibles, etc, y tiene una velocidad de ejecución muy competitiva. Las principales ventajas de este lenguaje son:

- Dispone de todas las características propias de cualquier lenguaje orientado a objetos: encapsulación, herencia y polimorfismo,
- Permite definir estructuras, que son clases un tanto especiales: sus objetos se almacenan en pila de acceso rápido,
- Es un lenguaje fuertemente tipado, lo que significa que se controla que todas las conversiones entre tipos se realicen de forma compatible, lo que asegura que nunca se acceda fuera del espacio de memoria ocupado por un objeto. Así se evitan frecuentes errores de programación y se consigue que los programas no puedan poner en peligro la integridad de otras aplicaciones,



• Figura 2.2 Operaciones en el análisis de circuitos

- Incorpora propiedades que permiten el acceso controlado a miembros de una clase, y
- Permite la definición del significado de los operadores básicos del lenguaje (+, -, *, &, =, etc.) para nuestros propios tipos de datos, lo que facilita enormemente tanto la legibilidad de las aplicaciones como el esfuerzo necesario para escribirlas.

MODELOS EN RED

Capítulo 3

El diseño de modelos en red, que constituye una de las etapas de aplicación del método de redes, descansa en el la analogía eléctrica de cualquier proceso físico. Se trata de construir un circuito eléctrico (modelo en red) cuyas ecuaciones son formalmente equivalentes a las del problema que se estudia. Cuando el modelo está bien construido, podemos disponer de su solución haciéndolo correr en un programa adecuado de simulación de circuitos. Estos programas disponen en la actualidad de algoritmos tan potentes que son capaces de proporcionar la solución exacta del circuito con errores prácticamente nulos; esto hace que 'sdi el circuito o modelo está bien diseñado la solución es completamente fiable. En los problemas que nos ocupan, no es necesario realizar una reticulación espacial lo que refuerza aún más la afirmación anterior ya que los errores debidos al tamaño de la retícula espacial no afectan a la solución, quedando ésta afectada solamente por el error cometido en la discretización temporal (la elección del tiempo de paso o 'time step'). Para optimizar el tiempo de computación, este tiempo de paso es variable, ajustándose a los mayores o menores cambios que se producen en la evolución de los variables o parámetros del problema, y salvo indicación del usuario su valor máximo se ajusta para proporcionar soluciones precisas y fiables. El usuario puede, no obstante, reducir a voluntad el tipo de paso para aumentar la precisión de la solución, a costa de mayores tiempos de computación.

En relación con lo anterior, cabe mencionar que merced a la aplicación de estos códigos a problemas eléctricos o electrónicos en los que se trabaja con frecuencias muy elevadas y formas de onda fuertemente no lineales, los algoritmos de cálculo

matemático implementados en éstos son los más potentes y fiables. Además, los teoremas sobre circuitos referidos a la conservación de la corriente eléctrica o carga (1ª ley de Kirchhoff) y a la unicidad del potencial eléctrico (2ª ley de Kirchhoff), naturalmente implementados en los códigos por ser teoremas propios, ajustan continuamente los residuos resultantes de las operaciones, reduciéndolos a valores prácticamente nulos.

Los argumentos en los que descansa el diseño de modelos en red, aplicados en este capítulo, son básicos: i) por un lado, para la construcción de las sucesivas derivadas de las variables se utiliza un dispositivo electrónico lineal, el condensador, merced a la relación constituyente entre las variables tensión eléctrica en sus extremos y corriente instantánea que circula por el mismo; ésta es una relación de tipo diferencial; ii) por otro lado, la existencia de dispositivos especiales en las librerías de estos circuitos, las llamadas fuentes de corriente controladas, que permiten implementar en el modelo cualquier tipo de acoplamiento de las ecuaciones; éstas fuentes, además, son la base del diseño para obtener las sucesivas derivadas y el tiempo en sus respectivos circuitos auxiliares, y iii) la implementación de los términos de cada ecuación del problema como corrientes eléctricas de ramas que se balancean en un punto o nodo común. Estas tres reglas son suficientes para abordar cualquiera de los problemas planteados, por complejo que sea.

A continuación, se desarrolla en detalle, en primer lugar, las etapas seguidas para el diseño de los modelos en red (circuitos eléctricos) de ecuaciones diferenciales ordinarias, de cualquier orden y grado, lineales y no lineales, y sistemas acoplados de estas ecuaciones con cualquier número de variables dependientes. En segundo lugar se explica la estructura de los modelos en red en forma de archivo de texto (archivos de texto de extensión '.cir') que se elaboran a partir de los circuitos eléctricos. Estos modelos en red, en forma de archivo de texto, se ejecutan en el programa de simulación de circuitos Pspice [1994] o en su versión moderna Orcad proporcionando la solución numérica del problema.

Conviene mencionar que, en general, los programas de simulación de circuitos disponen de un entorno propio (llamado Schematics en los códigos Pspice y Orcad)

que permite el diseño o elaboración directa de modelos (circuitos eléctricos) usando la simbología normalizada propia de los componentes eléctricos, pero esta opción no es útil para elaborar un software de resolución de ecuaciones como el que perseguimos.

La versión de un modelo en red en forma de un archivo de texto puede realizarse con relativa facilidad mediante programación por lo que ésta es la opción que seguimos en el presente proyecto usando Matlab. Para ello nos hemos detenido en el estudio de la estructura más intuitiva y adecuada de estos archivos en relación con todos los aspectos del modelo determinan el mismo tales como

- i) Información sobre el enunciado o tipo de problema,
- ii) Nomenclatura de variables,
- iii) Nomenclatura de componentes eléctricos que componen el modelo (resistencias, condensadores, fuentes controladas y fuentes independientes, etc.),
- iv) Nomenclatura de nodos separando aquellos que pertenecen a circuitos principales y auxiliares, e identificando fácilmente aquéllos que contienen las soluciones,
- v) Estructura y orden de los circuitos asociados con las diferentes variables, separando los circuitos principales de los auxiliares y del circuito generador de la variable tiempo,
- vi) Organización de sentencias complementarias (listado de parámetros, ventanas temporales de simulación, tablas de los resultados...)

Los modelos en red solo están completos cuando se especifica un conjunto de sentencias complementarias referidas a parámetros de la simulación. Por ejemplo, listado de variables de salida cuya impresión queremos obtener mediante tablas; intervalo temporal de la impresión, tolerancia relativa en los cálculos numéricos del programa, ventanas temporales de simulación, etc. La explicación sobre el formato de estas sentencias y el conjunto necesario de ellas para proceder a una simulación concreta se incluye en el Capítulo 4, capítulo donde se explica el funcionamiento de programa CODENS.

3.1. Componentes eléctricos y ecuaciones constitutivas

Podemos clasificar los tipos de componentes eléctricos necesarios para implementar los diferentes términos de la ecuación o ecuaciones diferenciales cuyo modelo en red se pretende diseñar en tres clases:

- i) Fuentes independientes de corriente o tensión,
- ii) Elementos pasivos lineales (resistencias, condensadores y bobinas) y,
- iii) Fuentes controladas de tensión o corriente.

3.1.1. Fuentes independientes

Se trata de dispositivos que aportan energía al sistema. Las fuentes independientes de tensión o corriente son necesarias en general para implementar cierto tipo de condiciones iniciales mientras que las independientes son de uso menos frecuente. En las librerías de los programas Pspice y Orcad existe un extenso tipo de estas fuentes que incluyen diferentes formas de onda de salida: fuentes de valor constante, sinusoidales, rectangulares, en escalón, lineal a tramos, etc.

3.1.2. Elementos pasivos

Se trata de componentes eléctricos que no general potencia, bien la disipan (transformación energética) bien tienen la capacidad de almacenarla y/o entregarla a la red; constituyen lo que en el método de redes se llaman monopuertas pasivas. En función de la relación expresa existente entre las variables corriente y tensión en el elemento, éste tiene un nombre específico.

Resistencia

Es un elemento de circuito cuya relación entre la corriente que circula por el mismo, i_R (A), y la diferencia potencial en sus extremos, v_R (V), es de la forma

$$i_R = v_R/R$$

donde R es el valor de la resistencia (Ω). En el archivo de texto del modelo este componente empieza por la letra R .



Figura 3.1 Símbolo de la resistencia

La representación simbólica de una monopuerta resistiva se muestra en la Figura 3.1. La traducción al modelo en red es una resistencia eléctrica de valor R ohmios para el caso lineal o una fuente controlada de corriente o tensión para el caso no lineal

Condensador

Es un elemento de circuito asociado a una relación lineal entre la variable corriente eléctrica (i_C) y la derivada temporal de la variable diferencia de potencial eléctrico en sus extremos (v_C). Así,

$$i_C = C(dv_C/dt)$$

donde C es el valor de la capacidad del condensador en faradios. Su representación simbólica se muestra en la Figura 3.2. En el archivo de texto del modelo este componente empieza por la letra C .



Figura 3.2 Símbolo de un condensador

Inductor o bobina

Es un elemento de circuito asociado a una relación lineal entre la variable diferencia de potencial en sus extremos (v_L) y la derivada temporal de la corriente eléctrica a su través (i_L). Así,

$$i_L = L(di_L/dt)$$

donde L es el valor de la autoinducción en henrios. Su representación simbólica se muestra en la Figura 3.3. En el archivo de texto del modelo este componente empieza por la letra L .



Figura 3.3 Símbolo de una bobina

3.1.3. Fuentes controladas

Se trata de componentes especiales asociados a relaciones constitutivas entre variables conjugadas o no, expresadas mediante cualquier función que no contiene explícitamente el tiempo. Se trata de elementos de entradas múltiples con una única salida que corresponde a una corriente o una tensión que depende funcionalmente de valores de tensión y/o corriente en otras partes del circuito, nudos o componentes respectivamente. Estas fuentes van a permitir especificar acoplamientos o cualquier otro tipo de no linealidad contenido en la ecuación o sistema de ecuaciones que se pretende simular.

Existen cuatro tipos de fuentes controladas, cuyos símbolos se muestran en la Figura 3.4:

- i) Fuentes de tensión controladas por tensión (E),
- ii) Fuentes de tensión controladas por corriente (H),
- iii) Fuentes de corriente controladas por tensión (G),
- iv) Fuentes de corriente controladas por corriente (F).

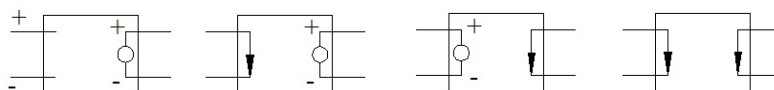


Figura 3.4 Representación de los cuatro tipos de fuentes controladas

E, H, G y F son las primeras letras con las que se designan estos componentes en el archivo de texto del modelo. La salida de una fuente E es una tensión que se expresa en función de otras tensiones (diferencias de potencial) entre cualquier par de nodos del circuito (o en los extremos de un componente resistivo, por ejemplo). La relación entre la tensión de salida de la fuente E y las tensiones del circuito de las que depende ésta es una función matemática arbitraria; las tensiones que constituyen los argumentos de esta expresión matemática son las variables de entrada de la fuente y suelen escribirse junto a la fuente entre corchetes.

Por otro lado, la salida de una fuente H es una tensión que se expresa en función de la corriente en un elemento cualquiera del circuito. Para facilitar la especificación de esta fuente la corriente del elemento que la controla se hace circular por una batería de

tensión nula en serie con el mismo (batería que hace las veces de galvanómetro); de esta forma el elemento de control de esta fuente es siempre el mismo. Existe una restricción en cuanto a que la relación entre la tensión de salida y la corriente de la batería es siempre lineal. La variable de control suele mostrarse también entre corchetes junto a la representación o símbolo de la fuente.

La fuente G proporciona una corriente de salida que depende de las tensiones existentes en cada instante entre pares de nodos o componentes del circuito. Esta dependencia es una función de cualquier tipo; las tensiones que constituyen los argumentos de esta expresión matemática son las variables de entrada de la fuente y se escriben junto a la fuente entre corchetes.

Por último, la fuente F proporciona una corriente de salida función de la corriente que circula por algún elemento del circuito. Al igual que para la fuente H, su especificación se facilita usando una batería de tensión nula en serie con el elemento cuya corriente es el argumento de control de esta fuente.

La especificación de estas fuentes en el archivo de texto del modelo se explica en el Capítulo 4. Aunque son posibles diferentes formas de especificación, la estructura que adoptaremos para la especificación (de acuerdo con los códigos Pspice y Orcad) es la siguiente: 'nombre de la fuente' seguido de los 'nodos de conexión' y seguido de 'valor de la tensión o corriente de salida'. En general, el valor de la salida se especifica mediante 'VALUE = {...}'; entre corchetes se muestra la función que define la variable de salida de la fuente (más adelante se podrá ilustrar la especificación de las diferentes fuentes mediante ejemplos).

Las posibles combinaciones o usos de estas fuentes dan un margen de posibilidades suficientemente grande como para resolver el diseño de un modelo en red mediante circuitos distintos. El potencial de éstas para implementar los modelos en red de sistemas fuertemente no lineales es inmenso ya que su uso permite imponer una variable de salida que es función de otras variables del circuito, sin influir en ellas. Así, cualquier tipo de término que pueda aparecer en una ecuación diferencial ordinaria puede implementarse mediante circuitos auxiliares a partir de estas fuentes.

3.2. Diseño de modelos básicos

3.2.1. Circuitos asociados a las sucesivas derivadas

Como veremos más tarde, para el caso de problemas formulados por una sola ecuación diferencial, la solución $y(t)$ de la misma es la tensión en el único nodo del circuito principal; si se trata de un sistema de ecuaciones acopladas, las soluciones $y_1(t)$, $y_2(t)$, $y_3(t)$... se encuentran en los únicos nodos de los circuitos principales de cada ecuación (cada ecuación, en este caso, se asocia a una variable, por lo que existen tantos circuitos principales como variables dependientes). Estos nudos tienen la siguiente denominación (ver el capítulo de la descripción del programa): N_p_y1 (para la variable y_1), N_p_y2 (para la variable y_2), N_p_y3 (para la variable y_3)... Así, digamos que disponemos de las soluciones $y_1(t)$, $y_2(t)$, $y_3(t)$... del problema a partir de las cuales elaboraremos las sucesivas derivadas de estas soluciones.

Ciñéndonos al caso de problemas formulados por una sola ecuación diferencial, la primera derivada se obtiene mediante los circuitos auxiliares de la Figura 3.5. La fuente de tensión controlada por tensión E_{1_y1} , de ganancia unidad, proporciona una tensión instantánea de salida del mismo valor que la tensión instantánea existente en su entrada y dado que la entrada es la tensión del nodo principal del circuito y_1 , es decir la tensión $y_1(t)$, la salida es asimismo $y_1(t)$. De acuerdo con la ecuación constitutiva de un condensador, la corriente que circula por C_{E1_y1} es su capacidad multiplicada por la derivada de la tensión en sus extremos y si $C=1$, el valor de esta corriente es $dy_1(t)/dt$, la primera derivada de la solución $y_1(t)$. La batería $V_{nula_E1_y1}$ de tensión nula no altera la solución del circuito pero permite que la corriente del condensador pase a su través y sirva de corriente de control del generador de tensión controlado por corriente H_{1_y1} (la batería no es más que un galvanómetro ideal que sirve para controlar la fuente H_{1_y1}). La salida de ésta es una tensión de valor numérico igual a la corriente de entrada (transconductancia unidad) y acoplándole una resistencia unidad (R_{uni_a-H1}) en sus extremos tenemos en el nudo N_{H1_y1} la solución $dy_1(t)/dt$ en forma de tensión.

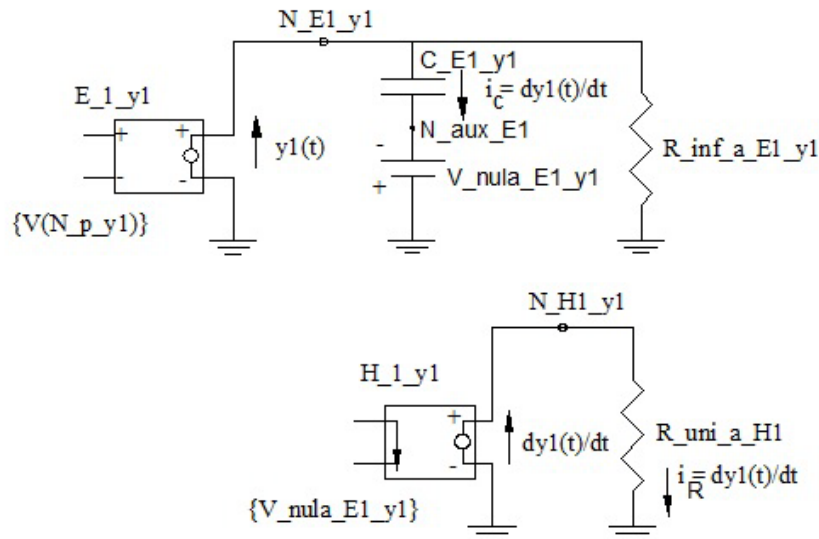


Figura 3.5 Circuitos auxiliares para la obtención de la primera derivada $dy_1(t)/dt$

Puede parecer que el circuito auxiliar formado por N_{H1_y1} y $R_{uni_a_H1}$ es innecesario y efectivamente lo es en tanto que la solución $dy_1(t)/dt$ que pretendemos obtener con los circuitos auxiliares de la Figura 3.5 ya se ha obtenido como corriente del condensador C_{E1_y1} ; es cierto, sin embargo, nuestro objetivo es obtener las sucesivas soluciones $dy_1(t)/dt$, $d^2y_1(t)/dt^2$, $d^3y_1(t)/dt^3$... como magnitudes de tensión (y no de corriente) pues van a ser tensiones, como más tarde se verá, las magnitudes que controlan las fuentes del circuito principal. Con ello, aunque los circuitos puedan parecer innecesariamente algo más complejos, se consigue una nomenclatura muy uniforme y regular en los componentes del modelo y el usuario puede seguir unos criterios ordenados en la interpretación del modelo. Por otro lado, la implementación de una resistencia infinita en paralelo con la fuente E_{1_y1} es un requisito del código Pspice para hacer las comprobaciones de continuidad en corriente continua del modelo con objeto de subsanar posible fallos (nudos o componentes que queden aislados por error en el diseño); también cabe pensar que la resistencia $R_{uni_a_H1}$, en paralelo con la fuente H_{1_y1} tampoco es necesaria ya que la solución buscada, $dy_1(t)/dt$, ya existe en forma de tensión en los extremos de la fuente; de nuevo esta resistencia (que podría ser también de valor infinito) es necesaria para

satisfacer los requisitos de continuidad impuestos por el código. Al elegir valor unidad para esta resistencia la corriente que circula por la misma tiene el mismo valor numérico que la tensión en sus extremos.

El circuito auxiliar correspondiente a la obtención de la segunda derivada $d^2y_1(t)/dt^2$, Figura 3.6, tiene una estructura similar al de la figura anterior. La tensión de control de la fuente de tensión E_2_y1 (de ganancia unidad) es ahora $dy_1(t)/dt$, y su valor se lee en el nudo N_H1_y1 de circuito auxiliar asociado a la obtención de la primera derivada (Figura 3.5); así, la tensión instantánea de salida de la fuente es del mismo valor que la tensión instantánea existente en su entrada, a saber, la primera derivada de la solución $y_1(t)$. En consecuencia, la corriente del condensador C_E1_y1 (de capacidad unidad) es la derivada de la tensión en sus extremos, $d^2y_1(t)/dt^2$. Al igual que anteriormente, la batería $V_nula_E2_y1$ de tensión nula recoge y lee esta corriente para controlar el generador de tensión H_2_y1 de transconductancia unidad, encargado de proporcionar el valor de la segunda derivada de y_1 , en sus extremos, en forma de tensión. Las resistencias del nuevo circuito auxiliar, $R_inf_a_E2_y1$ y $R_uni_a_H2$ cumplen el mismo papel que el ya mencionado en el circuito auxiliar de la Figura 3.5.

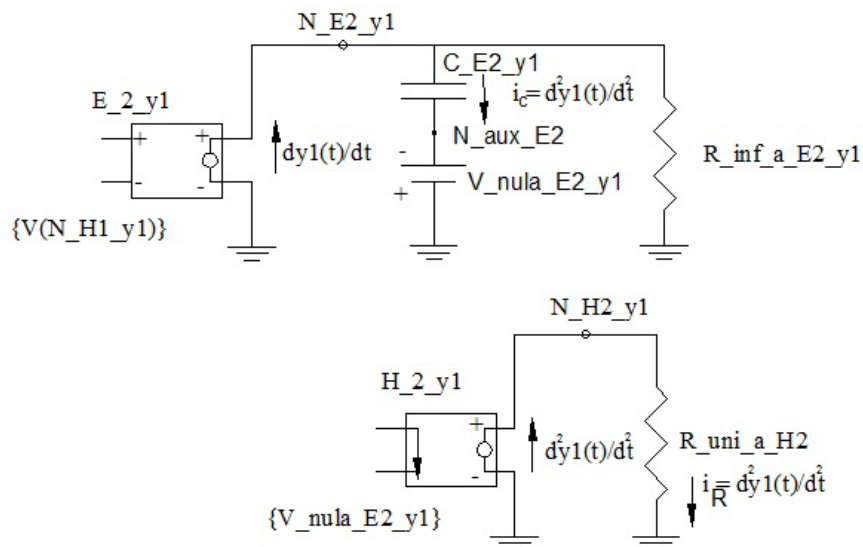


Figura 3.6 Circuitos auxiliares para la obtención de la segunda derivada $d^2y_1(t)/dt^2$

Las sucesivas (siguientes) derivadas de y_1 , $d^3y_1(t)/dt^3$, $d^4y_1(t)/dt^4$..., hasta la derivada correspondiente al orden de la ecuación, se obtienen mediante circuitos auxiliares similares a los de la Figuras 3.5 y 3.6 anteriores. El conjunto de circuitos auxiliares correspondientes a una ecuación dada puede simplificarse ya que las tensiones de salida de las fuentes E_{2_y1} , E_{3_y1} y sucesivas, ya existen en las salidas de las fuentes H_{1_y1} , H_{2_y1} y sucesivas, respectivamente, de los circuitos auxiliares anteriores. De esta forma puede combinarse cada fuente E_{n_y1} con la fuente H_{n-1_y1} en la forma indicada en la Figura 3.7. En esta figura, el circuito auxiliar de la izquierda, formado por las fuentes E_{n_y1} y H_{n-1_y1} es equivalente al circuito más simple de la derecha formado por la única fuente H_{n-1_y1} y sus componentes asociados.

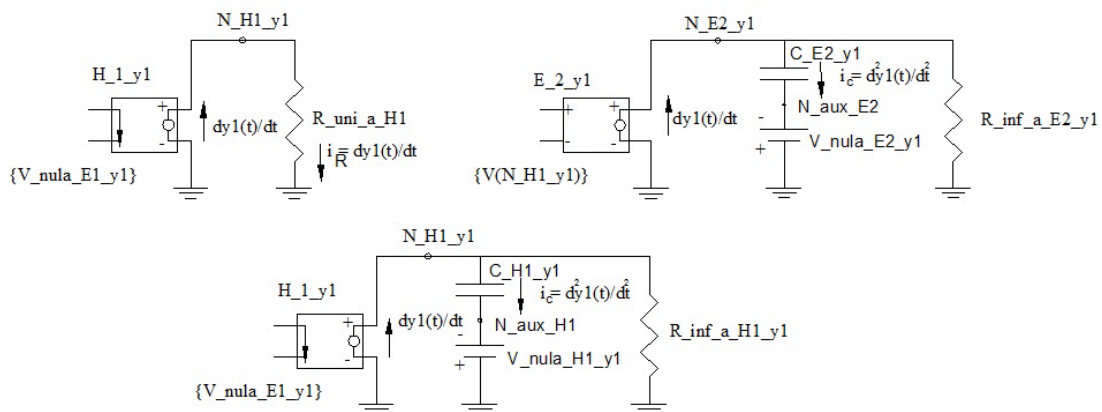


Figura 3.7 Circuitos auxiliares (arriba) y su simplificación (abajo)

Con todo, los circuitos auxiliares de las Figuras 3.5 y 3.6 pueden sustituirse por el circuito equivalente de la Figura 3.8, que tiene menos componentes y, en consecuencia, requiere menor tiempo de computación para su solución numérica. Con esta simplificación, las fuentes E_{2_y1} , E_{3_y1} ... desaparecen, al igual que las resistencias $R_{uni_a_H1}$, $R_{uni_a_H2}$..., excepto la resistencia unidad de la última fuente $H_{uni_a_Hn}$ asociada al orden n de la ecuación diferencial.

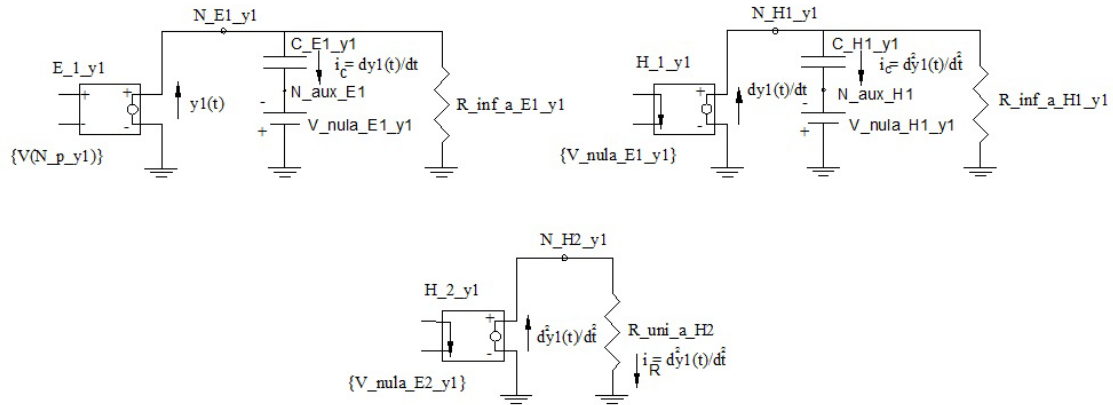


Figura 3.8 Circuitos auxiliares simplificados para la obtención de la primera y segunda derivada

Para aclarar en lo posible las anteriores reglas de diseño de los circuitos auxiliares, la Figura 3.9 muestra los correspondientes a una ecuación de tercer orden, ecuación que precisa de tres circuitos auxiliares más uno ‘de cierre’ para la obtención de la última (tercera) derivada en forma de tensión. La asociación entre soluciones y nodos para una ecuación de cuarto orden se muestra en el Cuadro 3.1, donde se incluye también las soluciones de las derivadas en forma de corrientes.

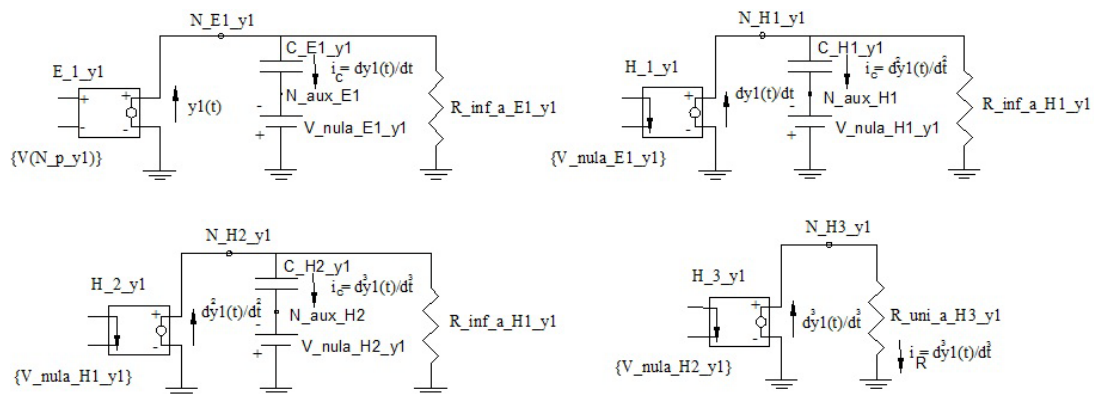


Figura 3.9 Circuitos auxiliares de una ecuación de tercer orden

	Solución en forma de tensión en el Nodo	Solución en forma de corriente en el condensador (o batería)
dy_1/dt	N_H1_y1 (también en N_E1_y1)	C_E1_y1 (V_nula_E1_y1)
d^2y_1/dt^2	N_H2_y1	C_H1_y1 (V_nula_H1_y1)
d^3y_1/dt^3	N_H3_y1	C_H2_y1 (V_nula_H2_y1)
d^4y_1/dt^4	N_H4_y1	C_H3_y1 (V_nula_H3_y1) (R_uni_a_H4_y1

Cuadro 3.1 Asociación entre soluciones derivadas, tensiones en nudos y corrientes eléctricas en componentes del modelo

La generalización del diseño de circuitos auxiliares a sistemas de N ecuaciones diferenciales de cualquier orden y grado es inmediata. Para cualquier incógnita y_i ($1 \leq i \leq N$) consideramos su ecuación asociada aquella en donde la incógnita tiene un orden mayor; caso de que una ecuación contenga dos incógnitas de un mismo orden y éste sea el orden mayor con el que dichas incógnitas aparecen en el sistema de ecuaciones, se opta por asociar dicha ecuación a cualquiera de las incógnitas y buscar otra ecuación de entre las restantes del sistema (aquella en la que la otra incógnita tenga un grado mayor) para asociarla a la otra incógnita.

Para ilustrar el diseño de los circuitos asociados a sistemas de ecuaciones, la Figura 3.10 muestra los circuitos auxiliares de la variable y_n ($n=1,2,3,\dots$) cuando ésta aparece con un orden máximo de 2 en el sistema de ecuaciones.

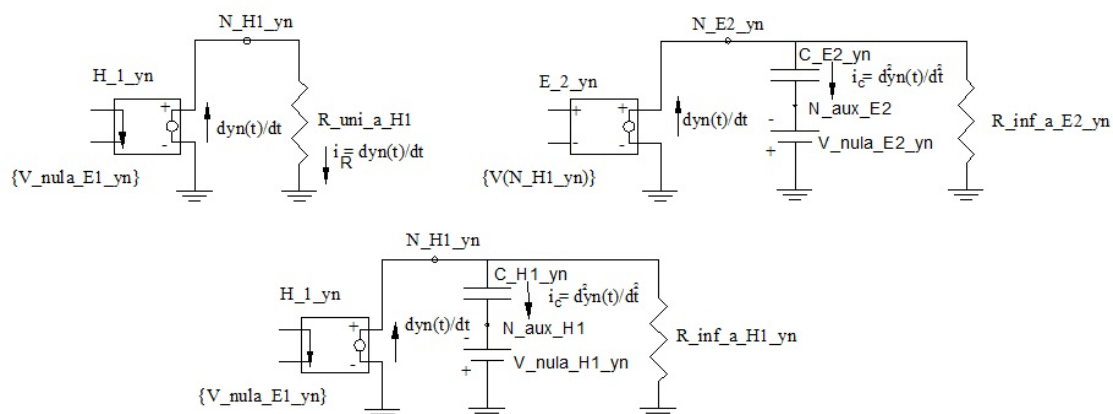


Figura 3.10 Circuitos auxiliares de la variable y_n (de segundo orden en el sistema)

El Cuadro 3.2 generaliza lo ilustrado en el cuadro anterior (asociación entre soluciones y nodos) para un orden máximo de 4. Adviértase que el número de derivadas asociadas a cada variable es el del orden de dicha variable en el sistema de ecuaciones (otras derivadas de orden mayor podrían eventualmente obtenerse, añadiendo los correspondientes circuitos auxiliares, no necesarios para la obtención de la solución de la variable asociada).

	Solución en forma de tensión en el Nodo	Solución en forma de corriente en el condensador (o batería)
dyn/dt	N_H1_yn (también en N_E1_yn)	C_E1_yn (V_nula_E1_yn)
d^2yn/dt^2	N_H2_yn	C_H1_yn (V_nula_H1_yn)
d^3yn/dt^3	N_H3_yn	C_H2_yn (V_nula_H2_yn)
d^4yn/dt^4	N_H4_yn	C_H3_yn (V_nula_H3_yn) (R_uni_a_H4_yn)

Cuadro 3.1 Asociación entre soluciones derivadas, tensiones en nudos y corrientes eléctricas en componentes del modelo

3.2.2. Circuitos asociados al tiempo y a los términos independientes

Consideramos como términos independientes de la ecuación diferencial todos aquellos que no contienen derivadas de las variables dependientes. Así, se trata de términos muy generales ya que pueden contener: i) funciones arbitrarias de las variables dependientes, ii) funciones de la variable tiempo y iii) combinaciones de i) y ii).

Como las soluciones de las variables dependientes $y_1(t)$, $y_2(t)$, $y_3(t)$... ya existen como tensiones en los nodos de los circuitos principales (que estudiaremos en el siguiente apartado), pueden utilizarse estas tensiones como argumentos para obtener el valor instantáneo de las funciones arbitrarias contenidas en los términos independientes. Completaremos la implementación de estos términos en el siguiente apartado.

En relación con la variable independiente tiempo, es inmediato obtener su valor instantáneo mediante el circuito de la Figura 11 formado por una fuente independiente del tipo PWL (piecewise linear) , V_time , y una resistencia de valor unidad, R_time . En el único nodo de este circuito, N_time , se mide el valor instantáneo de la variable t . La tensión de salida de esta fuente se expresa mediante programación mediante puntos que separan segmentos lineales. En realidad se trata de un tipo de fuente, contenida en las librerías de los códigos de simulación de circuitos, llamada fuente lineal a tramos, para cuya especificación se dan dos puntos con los mismos valores de abscisas y ordenadas: el origen (0,0) y el punto ($V_{m\acute{a}x}, t_{m\acute{a}x}$), con $V_{m\acute{a}x} = t_{m\acute{a}x}$. Con esto la salida es una línea recta de pendiente unidad que parte del origen y llega hasta un valor de tensión máxima equivalente al tiempo de simulación máxima que se desea simular, dato que introduce el usuario en el programa.

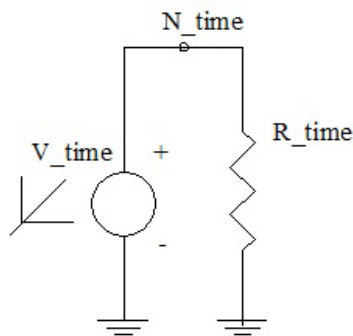


Figura 3.11. Circuito auxiliar para la obtención de la variable tiempo

Con todo, el valor de la variable independiente t se obtiene en el nudo V_time del circuito auxiliar de la figura anterior.

Ahora, sea cual sea la expresión del término independiente de cada ecuación del sistema, disponemos de la información necesaria para implementarlo: el valor instantáneo de cada variable en los nudos de los circuitos principales y el valor de t en el circuito auxiliar generador del tiempo; estas tensiones serán los posibles argumentos de las funciones (que pueden depender de uno de los anteriores argumentos o de más de uno de ellos) definen los términos independientes cuya implementación en el circuito principal se explica en el siguiente apartado. Por supuesto estos términos independientes pueden ser valores constantes.

En cualquier caso, sea cual sea la complejidad tanto de los términos independientes como de los coeficientes de los términos de la ecuación que multiplican a las expresiones derivadas de las variables dependientes (coeficientes que pueden ser también funciones de los argumentos y_1, y_2, \dots , del tiempo o de una mezcla de estas variables), o incluso existan términos formados por productos de expresiones derivadas de las variables dependientes, de cualquier orden o grado, siempre se puede recurrir al diseño de circuitos auxiliares que implementen la obtención de los valores buscados y resuelvan la implementación.

3.2.3. Circuito principal

Existen tantos circuitos principales como ecuaciones tiene el sistema y cada uno de ellos está asociado a una variable concreta para lo que se ha seguido la regla, anteriormente mencionada, de atribuir a cada circuito principal la variable de mayor orden que contenga (con los matices explicados en el apartado anterior para los casos especiales de ecuaciones que contengan dos o más variables del mismo orden).

Los circuitos principales implementan la topología de estas ecuaciones consideradas como balances entre sus términos. Cada término o miembro de la ecuación se constituye en una corriente que se balancea 'automáticamente' con las

correspondientes al resto de los términos en un nodo común cuya tensión es el valor de la variable dependiente asociada a dicha ecuación. Es decir, forzamos un balance en cada circuito principal cuyo cumplimiento impone las condiciones establecidas por el mismo entre las cuales está la definición de su ‘correspondiente’ variable en el nudo principal del circuito; la solución obtenida con la simulación no cabe sino que sea el valor buscado de dicha variable. Dicho de otra forma, en el circuito principal se imponen ciertas condiciones entre las corrientes (el propio balance en forma de teorema de Kirchhoff para las corrientes eléctricas) pero queda indeterminada la tensión en su único nudo, tensión que se proporciona en la simulación para las condiciones impuestas. Es lo dual de lo que ocurre en un circuito serie cuya solución, la corriente, se ajusta para satisfacer los requisitos impuestos por los valores de los potenciales conocidos de sus elementos.

Diseñemos en primer lugar el circuito principal correspondiente a una sola ecuación diferencial. Cada uno de sus términos se implementa mediante un generador de corriente controlado por tensión (G), por lo que existirán tantos generadores como términos contenga la ecuación. La corriente de estos generadores puede especificarse mediante programación indicando la expresión o función que define cada término y usando para los argumentos de los que depende éste (derivadas de cualquier orden, variables dependientes que contenga y el tiempo) el valor de la tensión en los nudos (principales o auxiliares) en los que se ha definido el argumento. Lo mejor es poner un ejemplo sencillo.

Sea la ecuación de variable y_1

$$d^3(y_1)/dt^3 + \text{sen}(t)[d^2(y_1)/dt^2] - (y_1)^2[d(y_1)/dt] + 3t^2(y_1)^{0.5} = 0 \quad (3.1)$$

Cada sumando está asociado a una fuente G. De acuerdo con la nomenclatura de estas fuentes descrita en el Capítulo 4, la asociación entre fuentes y términos es la siguiente:

$d^3(y_1)/dt^3$	\longleftrightarrow	G_3_y1
$\text{sen}(t)[d^2(y_1)/dt^2]$	\longleftrightarrow	G_2_y1
$(y_1)^2[d(y_1)/dt]$	\longleftrightarrow	G_1_y1
$3t^2(y_1)^{0.5}$	\longleftrightarrow	G_i_y1

De acuerdo con lo indicado en los epígrafes 3.2.1 y 3.2.2 deben implementarse los circuitos auxiliares necesarios para obtener las derivadas $d(y1)/dt$, $d^2(y1)/dt^2$, $d^3(y1)/dt^3$, y el circuito auxiliar que proporciona la variable tiempo. Las soluciones están en los nudos siguientes

$y1$	\longleftrightarrow	N_p_y1
$d(y1)/dt$	\longleftrightarrow	N_H1_y1
$d^2(y1)/dt^2$	\longleftrightarrow	N_H2_y1
$d^3(y1)/dt^3$	\longleftrightarrow	N_H3_y1
t	\longleftrightarrow	N_t

Con todo el circuito principal se muestra en la Figura 3.12. Puede observarse que la disposición del generador G_1_y1 está invertida (la corriente circula al revés que en los otros generadores) merced al signo de su correspondiente término en la ecuación diferencial. Entre corchetes se muestran los nudos correspondientes a las variables de control de cada fuente.

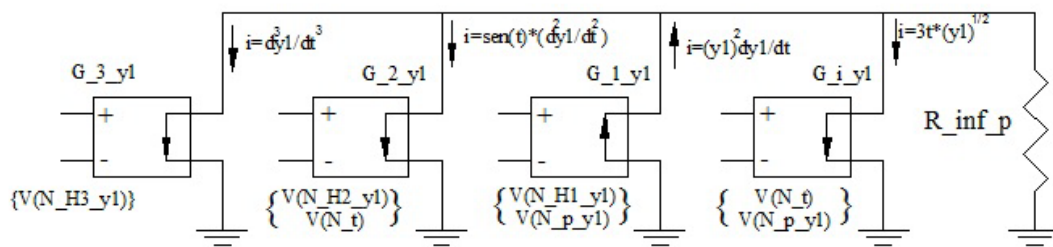


Figura 3.12 Circuito principal del ejemplo, (ecuación 3.1)

La corriente de salida de cada fuente se especifica mediante programación en el archivo de texto del modelo en red; es detalle de esta especificación se indica en el Capítulo 4 si bien escribimos, a continuación, la sentencia que indica la corriente de cada fuente:

G_3_y1	\longleftrightarrow	VALUE = {V(N_H3_y1)}
G_2_y1	\longleftrightarrow	VALUE = {(sin(V(N_t)))*V(N_H2_y1)}
G_1_y1	\longleftrightarrow	VALUE = {- ((V(N_p_y1))^2)*V(N_H1_y1)}
G_i_y1	\longleftrightarrow	VALUE = {3*((V(N_t))^2)*((V(N_p_y1))^2)}

Advierta el signo negativo de la corriente de G-1_y1 de acuerdo con la ecuación (3.1).
Advierta también que se ha escrito 'sin' para indicar la función seno en lugar de 'sen' ya que el código CODENS se ha realizado en lengua inglesa.

Un segundo ejemplo ilustrativo lo constituye la ecuación diferencial

$$0.1([d^4(y1)/dt^4]^2) + [d^3(y1)/dt^3]^2 + \sin(y1)[d^2(y1)/dt^2] - (1+(y1)^2)[d(y1)/dt] + 3\sin(t) + 0.2 = 0 \quad (3.2)$$

La asociación entre fuentes y términos es:

$0.1([d^2(y1)/dt^2]^2)$	\longleftrightarrow	G_4_y1
$[d^2(y1)/dt^2]^2$	\longleftrightarrow	G_3_y1
$\sin(y1)[d^2(y1)/dt^2]$	\longleftrightarrow	G_2_y1
$(1+(y1)^2)[d(y1)/dt]$	\longleftrightarrow	G_1_y1
$3\sin(t) + 0.2$	\longleftrightarrow	G_i_y1

Una vez implementados los circuitos auxiliares necesarios para obtener las derivadas $d(y1)/dt$, $d^2(y1)/dt^2$, $d^3(y1)/dt^3$ y $d^4(y1)/dt^4$ y el tiempo. Las soluciones están en los nudos siguientes

y1	\longleftrightarrow	N_p_y1
$d(y1)/dt$	\longleftrightarrow	N_H1_y1
$d^2(y1)/dt^2$	\longleftrightarrow	N_H2_y1
$d^3(y1)/dt^3$	\longleftrightarrow	N_H3_y1
$d^4(y1)/dt^4$	\longleftrightarrow	N_H4_y1
t	\longleftrightarrow	N_t

El circuito principal se muestra en la Figura 3.13.

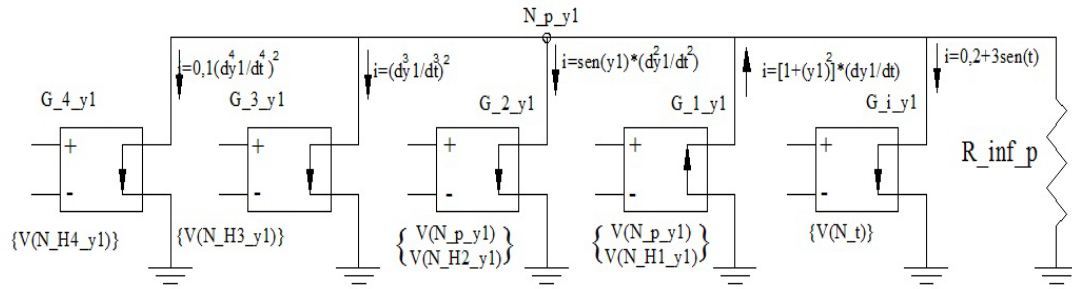


Figura 3.13 Circuito principal del ejemplo, (ecuación 3.1)

Las sentencias que indican las corrientes de cada fuente son:

G_4_y1	↔	VALUE = {0.1*(V(N_H4_y1)^2)}	
G_3_y1	↔	VALUE = {(V(N_H3_y1)^2)}	
G_2_y1	↔	VALUE	=
{(sin(V(N_p_y1)))*V(N_H2_y1)}			
G_1_y1	↔	VALUE	=
{-(1+((V(N_p_y1))^2))*V(N_H1_y1)}			
G_i_y1	↔	VALUE = {3*sin(N_t)+0.2}	

Adviértase el enorme número de paréntesis necesario para especificar las sentencias.

Pasamos a ilustrar un par de ejemplos de problemas de ecuaciones acopladas. El primero de ellos está enunciado por el conjunto de ecuaciones

$$3[d^2(y1)/dt^2] - 0.1[d(y1)/dt][d(y2)/dt] + 3+(y1)+0.5(y2) = 0 \quad (3.3a)$$

$$[d^2(y2)/dt^2] + (y2)[d(y1)/dt] + 0.3(y1) = 0 \quad (3.3b)$$

La asociación entre fuentes y términos es, para la primera ecuación (asociada a la variable y1):

$3[d^2(y1)/dt^2]$	↔	G_2_y1
$- 0.1[d(y1)/dt][d(y2)/dt]$	↔	G_1_y1
$3+(y1)+0.5(y2)$	↔	G_i_y1

y para la segunda ecuación (asociada a la variable y2):

$[d^2(y_2)/dt^2]$	\longleftrightarrow	G_{2_y1}
$(y_2)[d(y_1)/dt]$	\longleftrightarrow	G_{i1_y1}
$0.3(y_1)$	\longleftrightarrow	G_{i2_y1}

Las soluciones de las funciones y_1 e y_2 y sus derivadas están en los nudos

y_1	\longleftrightarrow	N_p_y1
$d(y_1)/dt$	\longleftrightarrow	N_H1_y1
$d^2(y_1)/dt^2$	\longleftrightarrow	N_H2_y1
t	\longleftrightarrow	N_t
y_2	\longleftrightarrow	N_p_y2
$d(y_2)/dt$	\longleftrightarrow	N_H1_y2
$d^2(y_2)/dt^2$	\longleftrightarrow	N_H2_y2
t	\longleftrightarrow	N_t

Los circuitos principales de cada variable se muestran en la Figura 3.14.

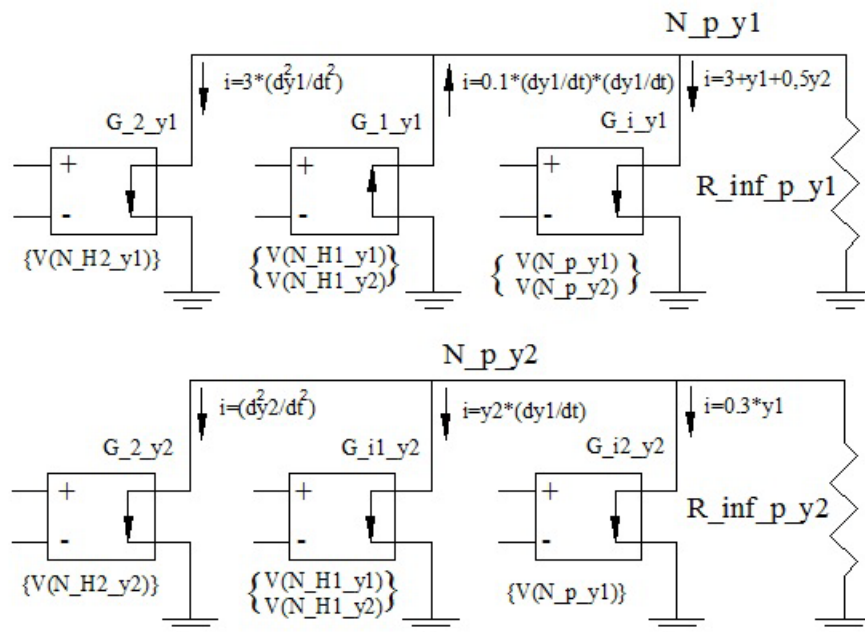


Figura 3.14 Circuitos principales del primer caso de ecuaciones acopladas (ecuaciones 3.3a y 3.3b)

Las sentencias que indican las corrientes de cada fuente son:

G_2_y1	↔	VALUE = {3*V(N_H2_y1)}	
G_i1_y1	↔	VALUE	= {-
0.1*V(N_H1_y1)*V(N_H1_y1)}			
G_i2_y1	↔	VALUE	=
{3+V(N_p_y1)+0.5*V(N_p_y2)}			
G_2_y2	↔	VALUE = {V(N_H2_y2)}	
G_1_y2	↔	VALUE = {V(N_p_y2)*V(N_H1_y1)}	
G_i_y2	↔	VALUE = {0.3*V(N_p_y1)}	

El segundo ejemplo lo constituye el conjunto de ecuaciones

$$[d^4(y1)/dt^4] - 3(y1)(y3)[d(y1)/dt] - 4.5(y1)(y2) = 0$$

(3.4a)

$$[d^3(y2)/dt^3] - ((y1)^{0.5})(y2)(y3) = 0$$

(3.4b)

$$[d^3(y3)/dt^3] - [d^2(y2)/dt^2] + (y1)(y3)[d(y3)/dt] + (y2)(y3) = 0$$

(3.4c)

La asociación entre fuentes y términos es, para la primera ecuación (asociada a la variable y1):

$d^4(y1)/dt^4$	↔	G_4_y1
$- 3(y1)(y3)[d(y1)/dt]$	↔	G_1_y1
$- 4.5(y1)(y2)$	↔	G_i_y1

para la segunda ecuación (asociada a la variable y2):

$d^3(y2)/dt^3$	↔	G_3_y2
$- ((y1)^{0.5})(y2)(y3)$	↔	G_i_y2

y para la tercera ecuación (asociada a la variable y3):

$d^3(y3)/dt^3$	↔	G_3_y3
$(y1)(y3)[d(y3)/dt]$	↔	G_1_y3
$- d^2(y2)/dt^2$	↔	G_i1_y3
$(y2)(y3)$	↔	G_i2_y3

Las soluciones de las funciones y_1 e y_2 y sus derivadas están en los nudos

y_1	\longleftrightarrow	N_{p_y1}
$d(y_1)/dt$	\longleftrightarrow	N_{H1_y1}
$d^2(y_1)/dt^2$	\longleftrightarrow	N_{H2_y1}
$d^3(y_1)/dt^3$	\longleftrightarrow	N_{H3_y1}
$d^4(y_1)/dt^4$	\longleftrightarrow	N_{H4_y1}
y_2	\longleftrightarrow	N_{p_y2}
$d(y_2)/dt$	\longleftrightarrow	N_{H1_y2}
$d^2(y_2)/dt^2$	\longleftrightarrow	N_{H2_y2}
$d^3(y_2)/dt^3$	\longleftrightarrow	N_{H3_y2}
y_3	\longleftrightarrow	N_{p_y2}
$d(y_3)/dt$	\longleftrightarrow	N_{H1_y3}
$d^2(y_3)/dt^2$	\longleftrightarrow	N_{H2_y3}
$d^3(y_3)/dt^3$	\longleftrightarrow	N_{H3_y3}

Los circuitos principales de cada variable se muestran en la Figura 3.15.

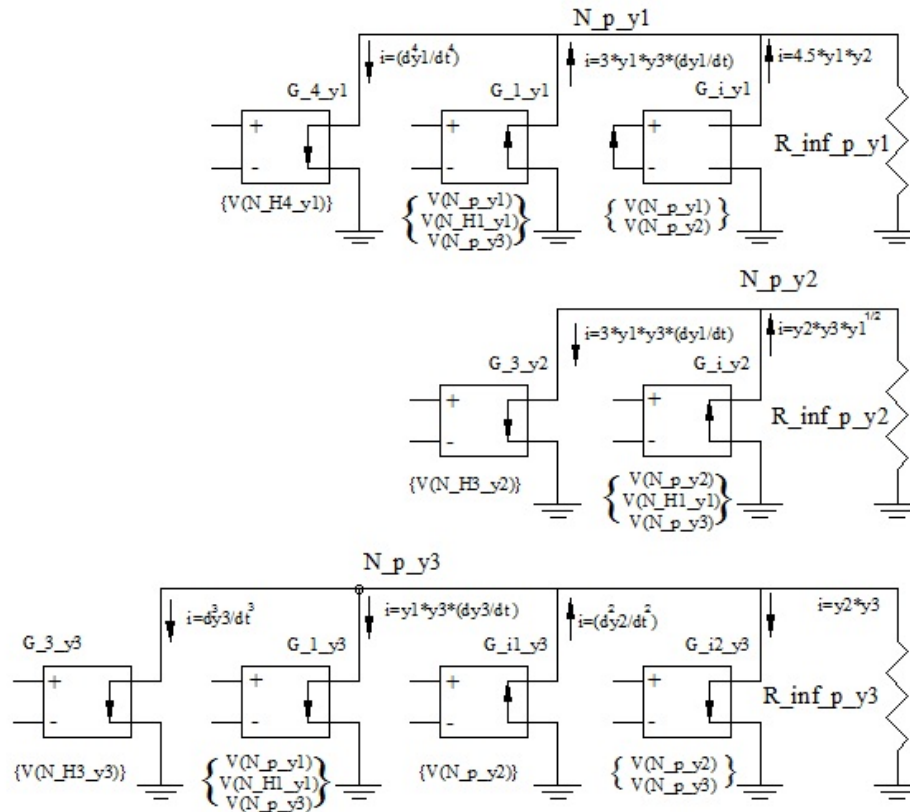


Figura 3.15 Circuitos principales del segundo ejemplo de ecuaciones acopladas
(ecuaciones 3.4a-c)

Las sentencias que indican las corrientes de cada fuente son:

G_{4_y1}	\longleftrightarrow	VALUE = $\{V(N_{H4_y1})\}$	
G_{1_y1}	\longleftrightarrow	VALUE	=
$\{3 * V(N_{p_y1}) * V(N_{p_y3}) * V(N_{H1_y1})\}$			
G_{i_y1}	\longleftrightarrow	VALUE	=
$\{4.5 * V(N_{p_y1}) * V(N_{p_y2})\}$			
G_{3_y2}	\longleftrightarrow	VALUE = $\{V(N_{H3_y2})\}$	
G_{i_y2}	\longleftrightarrow	VALUE = $\{((V(N_{p_y1}))^2 * V(N_{p_y2}) * V(N_{p_y3}))\}$	
G_{3_y3}	\longleftrightarrow	VALUE = $\{V(N_{H3_y3})\}$	
G_{2_y3}	\longleftrightarrow	VALUE = $\{V(N_{H2_y3})\}$	
G_{1_y3}	\longleftrightarrow	VALUE = $\{(V(N_{p_y1})) * (V(N_{p_y3})) * V(N_{H1_y3})\}$	

$$G_i_y3 \quad \longleftrightarrow \quad \text{VALUE} = \{V(N_p_y2)*V(N_p_y3)\}$$

Para terminar hay que hablar de las condiciones iniciales. Éstas se implementan como tensión en los condensadores respectivos. Así, la condición inicial $y1_{(t=0)} = c_{y1}$ se implementa como tensión inicial en el condensador C_E1_y1, la condición inicial $d(y1)/dt_{(t=0)} = c_{d(y1)/dt}$ se implementa como tensión inicial en el condensador C_H1_y1, la condición inicial $d^2(y1)/dt^2_{(t=0)} = c_{d^2(y1)/dt^2}$ se implementa como tensión inicial en el condensador C_H2_y1, y así sucesivamente. Y lo mismo para las condiciones iniciales del resto de las variables $y2, y3, \dots$

3.2.4. Modelos duales mediante bobinas

Aunque no es un objetivo de esta memoria queremos apuntar que los modelos anteriores pueden diseñarse utilizando bobinas de inducción en lugar de condensadores. La ecuación constitutiva de una bobina, o relación entre la corriente variable que circula por la misma y la diferencia de potencial entre sus extremos, viene dada por

$$v_L = L(di_L/dt)$$

donde L es el valor del coeficiente de inducción en henrios. Su representación simbólica se muestra en la Figura 3.3. En el archivo de texto del modelo este componente empieza por la letra L.

Las variables dependientes $y1, y2, y3, \dots$ se asocian a las corrientes (incógnitas) de los circuitos principales, formados por un solo lazo o malla, en los que cada corriente se ajusta para establecer la unicidad de los potenciales a lo largo de la malla (primer teorema de Kirchhoff). La corriente del circuito principal de la variable $y1$, que se lee mediante un galvanómetro o batería de tensión nula, sirve de control para el generador de corriente controlado por corriente del primer circuito auxiliar: en los extremos de dicho generador se conecta una bobina de autoinducción unidad con lo que la tensión en sus extremos es la primera derivada $d(y1)/dt$. Haciendo que esta tensión controle un generador de corriente controlado (segundo circuito auxiliar) en

suyos extremos hay una nueva bobina de autoinducción unidad, obtenemos la segunda derivada en sus extremos. Con este procedimiento implementamos todos los circuitos auxiliares necesarios para obtener las sucesivas derivadas.

El circuito principal está formado por generadores de tensión controlados por tensión, para implementar tanto los términos de la ecuación que contienen derivadas como los términos independientes (aunque éstos pueden implementarse con elementos más simples. Finalmente, las condiciones iniciales se implementan como corrientes iniciales en las bobinas.

EL PROGRAMA CODENS_14

Capítulo 4

El capítulo presente describe el programa CODENS_14 cuya elaboración es el objeto del presente proyecto. CODENS es el acrónimo de “CoupledOrdinaryDifferentialEquationby Network Simulation” y 14 se refiere al año corriente (2014).

Mediante MATLAB se han desarrollado los procesos de:

1. Elaboración del entorno de comunicación con el usuario, a través de ventanas.
2. Creación del modelo en red, basado en el método de simulación por redes, y su transformación en un archivo de texto inteligible para el código de simulación de circuitos usado (Pspice u Orcad).
3. Arranque del software de cálculo numérico (el código de simulación anterior).
4. Programación de las rutinas de programación necesarias para la representación gráfica de los resultados de la simulación.

El programa CODENS_14 resuelve, en principio, ecuaciones diferenciales ordinarias de cualquier orden (hasta cuarto orden), grado y tipo de no linealidad, mediante un código de programación y cálculo completamente accesible al usuario, así como sistemas acoplados de estas ecuaciones que contengan hasta nueve variables (manteniendo el orden cuatro para las derivadas de cada variable) . En su elaboración se han seguido reglas lógicas e intuitivas que permitan al usuario familiarizarse con su aplicación con el menor esfuerzo. A pesar de todo, dado la variedad de problemas que resuelve, que obliga a manipular un elevado número de variables y términos de las ecuaciones, lo que a su vez implica definir asimismo un elevado número de

componentes y nodos en el modelo en red, ha sido necesario realizar un esfuerzo especial para buscar una nomenclatura general adecuada y que pudiera extenderse desde los problemas más simples hasta los casos de mayor orden y mayor número de variables.

Como se ha descrito en el capítulo anterior, el diseño de los modelos en red de cada problema sigue unas reglas generales que permite implementar idénticos componentes en el modelo para cada una de las derivadas (primera, segunda...) de cada variable, con independencia del tipo de problema, y denominaciones con una misma estructura para cada uno de los nodos en los que se encuentran las soluciones. Cuando una ecuación de orden 'n' no contiene algún término derivativo de orden inferior a n, el programa implementa, siguiendo siempre unas mismas reglas, los circuitos auxiliares que elaboran la derivada no existente en la ecuación; es decir, en cada problema se implementan todos los circuitos auxiliares asociados con el conjunto completo de las n derivadas de cada variable, un circuito auxiliar por cada derivada (exista ésta o no en la ecuación). De esta forma la estructura de los modelos en red creados es muy general y la nomenclatura usada permite identificar fácilmente los componentes y nodos del circuito y, por ende, localizar los nudos o elementos de circuito en los que se encuentran las variables y las sucesivas derivadas que constituyen la solución numérica del problema.

La estructura del archivo de texto del modelo se ha organizado, asimismo, lo más ordenadamente posible y su acceso, tras la introducción de datos, permite la comprobación y modificación del mismo. La simulación da acceso (abre) a Pspice para, en primer lugar, presentar las soluciones en forma gráfica o tabulada en el propio entorno del programa, y en segundo lugar, acceder al resto de información contenida en la solución y, en general, no listada (lectura de corrientes en todos los elementos y de voltajes en todos los nudos del modelo). Finalmente, las rutinas de tratamiento de datos permiten representar las soluciones en el entorno gráfico de Matlab.

Mención especial merece el trabajo que ha sido preciso realizar para integrar en estos problemas (al margen del uso de parámetros en general) la posibilidad de:

i) incluir como términos independientes, o como coeficientes de los términos definidos por las sucesivas derivadas, funciones dependientes de tiempo, y

ii) incluir términos de acoplamiento que contengan derivadas de otras variables diferentes de las que aparecen con el mayor orden en una ecuación (la variable contenida en la derivada de mayor orden es la variable asociada directamente a esa ecuación, de forma que existe una asociación unívoca entre variables y ecuaciones del problema). En el diseño del modelo, estos términos se tratan como términos independientes para no afectar a la propia estructura del diseño en relación con el diseño de los circuitos auxiliares (se implementan en primer lugar todos los circuitos auxiliares correspondientes a las derivadas de la variable principal, o variable asociada a la ecuación, y a continuación se diseñan los circuitos auxiliares necesarios para implementar el término o términos independientes).

Como idioma de comunicación con el usuario en el programa CODENS_14 se ha optado por el inglés con objeto de darle la máxima difusión.

4.1. Estructura de las ecuaciones a resolver

4.1.1 Ecuaciones diferenciales ordinarias

El tipo de ecuación diferencial independiente que resuelve CODENS_14 podemos escribirlo en la forma

$$\begin{aligned} & \left(\frac{d^n y}{dt^n}\right)^{a_n} + f_{n-1}(y, t) \left(\frac{d^{n-1} y}{dt^{n-1}}\right)^{a_{n-1}} + f_{n-2}(y, t) \left(\frac{d^{n-2} y}{dt^{n-2}}\right)^{a_{n-2}} + f_{n-3}(y, t) \left(\frac{d^{n-3} y}{dt^{n-3}}\right)^{a_{n-3}} \\ & + \dots \\ & + f_1(y, t) \left(\frac{dy}{dt}\right)^{a_1} + f_0(y, t)[y] + f_i(y, t) = 0 \end{aligned} \quad (1)$$

donde f es una función arbitraria de los argumentos señalados. Esta ecuación está constituida por $n+2$ términos que identificaremos por los símbolos entre comillas siguientes:

$$\left(\frac{d^n y}{dt^n}\right)^{a_n} \quad \text{"N"}$$

$$f_{n-1}(y, t) \left(\frac{d^{n-1}y}{dt^{n-1}} \right)^{a_{n-1}} \Leftrightarrow \text{"N-1"}$$

$$f_{n-2}(y, t) \left(\frac{d^{n-2}y}{dt^{n-2}} \right)^{a_{n-2}} \Leftrightarrow \text{"N-2"}$$

$$f_{n-3}(y, t) \left(\frac{d^{n-3}y}{dt^{n-3}} \right)^{a_{n-3}} \Leftrightarrow \text{"N-3"}$$

...

$$f_1(y, t) \left(\frac{dy}{dt} \right)^{a_1} \Leftrightarrow \text{"1"}$$

$$f_0(y, t)[y] \Leftrightarrow \text{"0"}$$

$$f_i(y, t) \Leftrightarrow \text{"i"}$$

Así, cada término o sumando de la ecuación se identifica con el orden en que aparece su derivada y el término independiente con "i". El término que contienen la función, que puede o no existir, se designa como "0" con independencia de que, en general, $f_0(y, t)$ sean también función (polinómica, por ejemplo) de y , lo que implica que el término "0", en definitiva, pueda tener como argumento expresiones potenciales de la variable en lugar de expresiones lineales de la misma. Por otro lado, el término independiente [i] puede estar constituido por funciones de y , de t o de ambas variables simultáneamente, o simplemente se trata de una constante. Lo mismo podemos decir de los coeficientes de los términos derivativos $f_{n-1}(y, t)$, $f_{n-2}(y, t)$, $f_{n-3}(y, t)$, ..., $f_1(y, t)$ y $f_0(y, t)$.

En cuanto a los exponentes, a_{n-1} , a_{n-2} , ... a_0 , estos pueden tomar valores arbitrarios. El coeficiente a_n vale la unidad en todos los casos de ecuaciones que representan procesos físicos por lo hemos eliminado la posibilidad de asignar un valor diferente de la unidad a este coeficiente. Si se pretende resolver un problema donde $a_n \neq 1$ habría que hacer un diseño de modelo en red particular, siguiendo las reglas del capítulo anterior. Es curioso observar, por cierto, que asignar a este coeficiente un valor distinto de la unidad produce problemas de convergencia, un tema que dejamos pendiente por el momento.

4.1.2 Sistemas acoplados de ecuaciones diferenciales ordinarias

Las ecuaciones que constituyen estos sistemas se pueden escribir en la forma

$$\begin{aligned} \left(\frac{d^n y_1}{dt^n}\right)^{a_{n,1}} + f_{n-1,1}(y_j, t) \left(\frac{d^{n-1} y_1}{dt^{n-1}}\right)^{a_{n-1,1}} + f_{n-2,1}(y_j, t) \left(\frac{d^{n-2} y_1}{dt^{n-2}}\right)^{a_{n-2,1}} + \dots \\ + f_{1,1}(y_j, t) \left(\frac{dy_1}{dt}\right)^{a_{1,1}} + f_{0,1}(y_j, t)[y_1] + f_{i,1}(y_j, t) = 0 \end{aligned} \quad (2)$$

$$\begin{aligned} \left(\frac{d^n y_2}{dt^n}\right)^{a_{n,2}} + f_{n-1,2}(y_j, t) \left(\frac{d^{n-1} y_2}{dt^{n-1}}\right)^{a_{n-1,2}} + f_{n-2,2}(y_j, t) \left(\frac{d^{n-2} y_2}{dt^{n-2}}\right)^{a_{n-2,2}} + \dots \\ + f_{1,2}(y_j, t) \left(\frac{dy_2}{dt}\right)^{a_{1,2}} + f_{0,2}(y_j, t)[y_2] + f_{i,2}(y_j, t) = 0 \end{aligned} \quad (3)$$

$$\begin{aligned} \left(\frac{d^n y_3}{dt^n}\right)^{a_{n,3}} + f_{n-1,3}(y_j, t) \left(\frac{d^{n-1} y_3}{dt^{n-1}}\right)^{a_{n-1,3}} + f_{n-2,3}(y_j, t) \left(\frac{d^{n-2} y_3}{dt^{n-2}}\right)^{a_{n-2,3}} + \dots \\ + f_{1,3}(y_j, t) \left(\frac{dy_3}{dt}\right)^{a_{1,3}} + f_{0,3}(y_j, t)[y_3] + f_{i,3}(y_j, t) = 0 \end{aligned} \quad (4)$$

... hasta ocho variables.

Aquí, f es una función arbitraria de los argumentos señalados en los que se incluyen variables diferentes que la propia de la ecuación (el subíndice j identifica la variable y puede haber más de una y_j). Se ve que cada ecuación está asociada a una variable, la del término de mayor orden (obviamente, no es preciso que todas las ecuaciones tengan el mismo orden n . Cada ecuación, con su propio orden n , contiene $n+2$ términos que identificaremos por los símbolos entre comillas de la siguiente forma:

$$\begin{aligned} \left(\frac{d^n y_1}{dt^n}\right)^{a_{n,1}} \Leftrightarrow \text{"N_y1"}; \quad \left(\frac{d^n y_2}{dt^n}\right)^{a_{n,2}} \Leftrightarrow \text{"N_y2"} \dots \\ f_{n-1,1}(y_j, t) \left(\frac{d^{n-1} y_1}{dt^{n-1}}\right)^{a_{n-1,1}} \Leftrightarrow \text{"N-1_y1"}; \quad f_{n-1,2}(y_j, t) \left(\frac{d^{n-1} y_2}{dt^{n-1}}\right)^{a_{n-1,2}} \Leftrightarrow \text{"N-1_y2"} \dots \end{aligned}$$

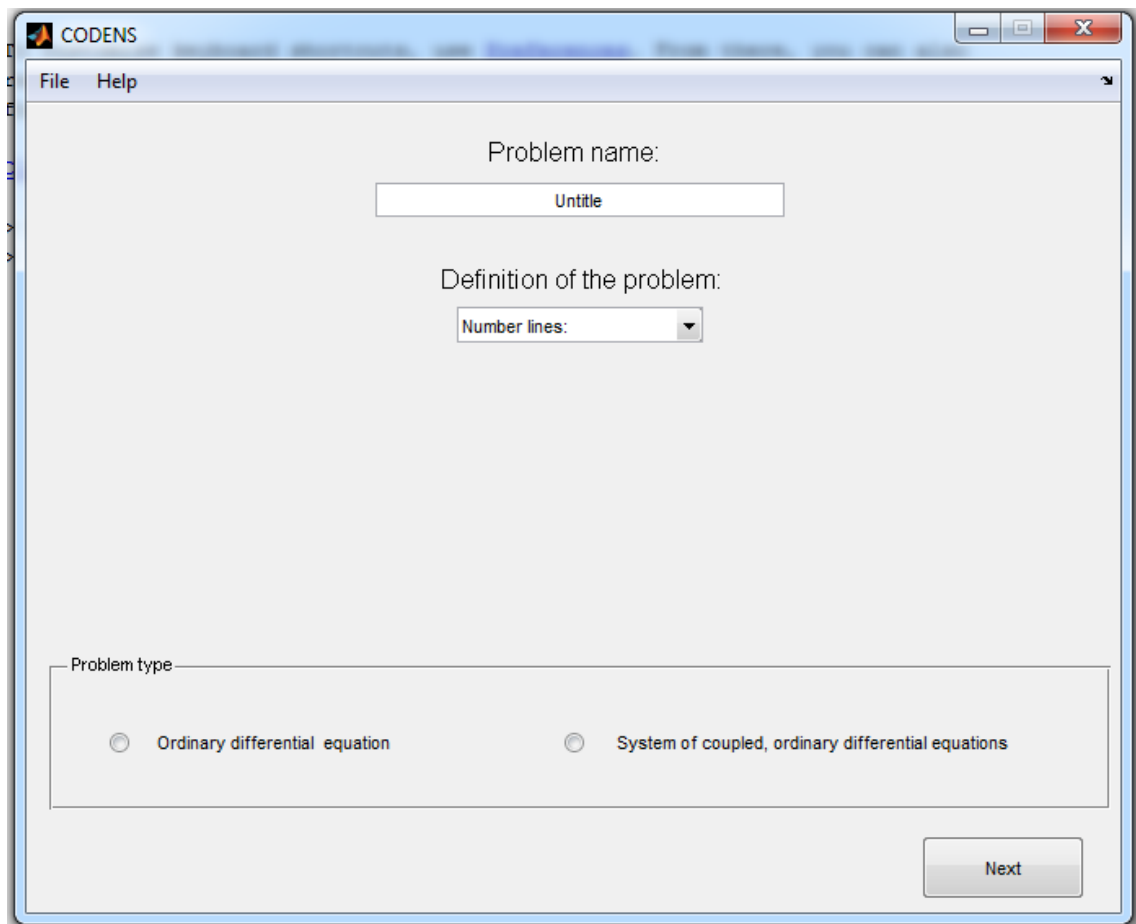
$$\begin{aligned}
f_{n-1,2}(y_j, t) \left(\frac{d^{n-2}y_1}{dt^{n-1}} \right)^{a_{n-1,3}} &\Leftrightarrow \text{"N-2_y1"}; & f_{n-1,2}(y_j, t) \left(\frac{d^{n-2}y_2}{dt^{n-1}} \right)^{a_{n-1,2}} &\Leftrightarrow \text{"N-2_y2"} \dots \\
&\dots & & \\
f_{1,1}(y_j, t) \frac{dy_1}{dt} &\Leftrightarrow \text{"1y1"} & f_{1,2}(y_j, t) \frac{dy_2}{dt} &\Leftrightarrow \text{"1y2"} \dots \\
f_{0,1}(y_j, t)[y_1] &\Leftrightarrow \text{"0y1"} & f_{0,2}(y_j, t)[y_2] &\Leftrightarrow \text{"0y2"} \dots \\
f_{i,1}(y_j, t) &\Leftrightarrow \text{"iy1"} & f_{i,2}(y_j, t) &\Leftrightarrow \text{"iy2"} \dots
\end{aligned}$$

Así cada término o sumando de una ecuación se identifica con el orden en que aparece su derivada (o "i" para el término independiente) y el nombre de la variable propia de la ecuación. Caben hacer las mismas consideraciones que en el caso de ecuaciones diferenciales individuales en relación con las funciones $f_0(y_1, t)$ y con los exponentes.

4.2 El Programa CODENS_14

4.2.1 Interfaz de CODENS_14

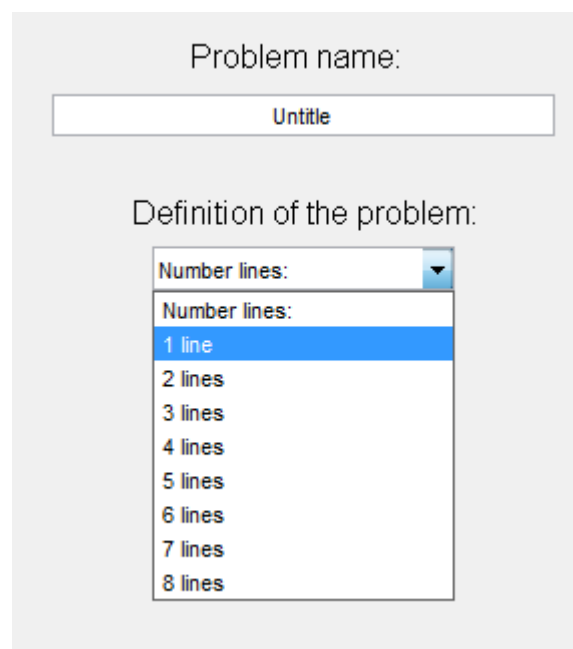
Cuando ejecutamos el programa aparecerá la pantalla mostrada en la figura 4.2. Se debe introducir el nombre del problema abordado ('Problemname') y una breve descripción del mismo ('Definition of theproblem'), a través de un conjunto seleccionado de líneas, que nos sirva de recordatorio de su contenido, figura 4.3.



The screenshot shows the CODENS application window. The title bar includes the CODENS logo and the text 'CODENS'. The menu bar has 'File' and 'Help'. The main area contains the following elements:

- 'Problem name:' label above a text box containing 'Untitled'.
- 'Definition of the problem:' label above a 'Number lines:' dropdown menu.
- 'Problem type:' label above a group box containing two radio buttons:
 - ☐ Ordinary differential equation
 - ☐ System of coupled, ordinary differential equations
- 'Next' button at the bottom right.

Figura 4.2 Pantalla inicio.



This image is a close-up of the 'Number lines:' dropdown menu from the previous figure. The dropdown is open, showing a list of options:

- Number lines: (header)
- 1 line
- 2 lines
- 3 lines
- 4 lines
- 5 lines
- 6 lines
- 7 lines
- 8 lines

Figura 4.3 Nombre y descripción del programa

Tanto el nombre como la descripción introducida aparecen al principio del archivo de texto (modelo en red) del problema, en tantas líneas de texto como número de ellas se haya seleccionado en la pantalla del programa. En esta pantalla también se introduce el tipo de problema, si es de una sola ecuación diferencial o de un sistema de ecuaciones acopladas. Como novedad, el programa no nos permitirá avanzar sin introducir un nombre para el problema. En el caso de no hacerlo, saldrá por defecto el nombre de “untitled”.

Tras la selección de ecuaciones simples o acopladas, accedemos a la siguiente pantalla mediante el botón ‘Next’, figura 4.4.

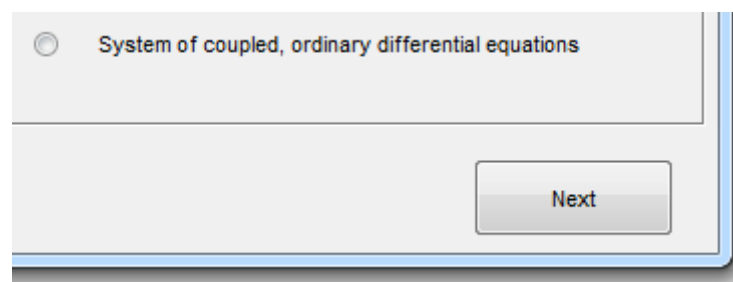


Figura 4.4 Botón Next, pantalla siguiente.

En la parte superior izquierda de la pantalla se incluyen los botones típicos ‘File’ y ‘Help’, figura 4.5. El primero de ellos tiene la opción ‘New’, que sirve para eliminar el nombre y/o descripción introducida en los cuadros de la pantalla, y ‘Close’ que sirve para cerrar el programa. El botón (‘Help’) o de ayuda no ha podido ser implementado en la versión CODENS_14, pero éste contendrá el manual de funcionamiento del programa y la lista de errores más comunes que aparecen por una inadecuada manipulación; estas opciones, así como el registro del programa como propiedad intelectual.

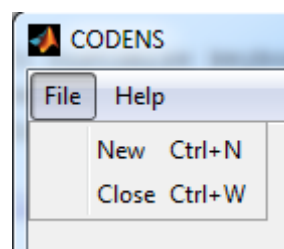


Figura 4.5 Botones de nuevo ejercicio y cerrar.

Si seleccionamos un ejercicio de ecuaciones diferenciales simples.

Al seleccionar 'Ordinarydifferentialequation' como se muestra en la pantalla de la figura 4.2, aparecerá una pantalla para la introducción de datos dando lugar a la figura 4.6. Aparece por defecto el formato típico de una ecuación diferencial normal, figura 4.6.1 y pide al usuario definir el grado de la ecuación a estudiar, 'Order of the equation', figura 4.6.2.

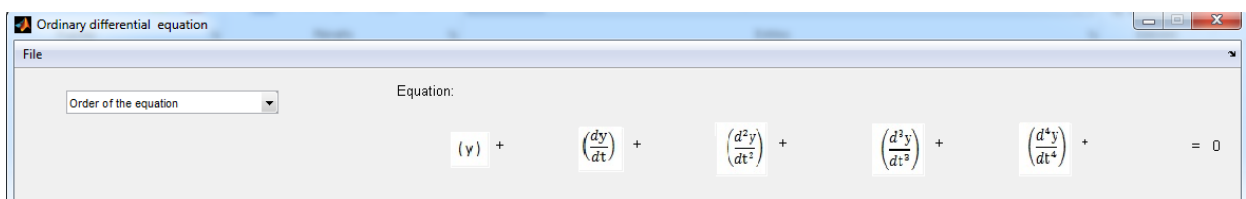


Figura 4.6 Introducción de datos para el problema 'Ordinarydifferentialequation'

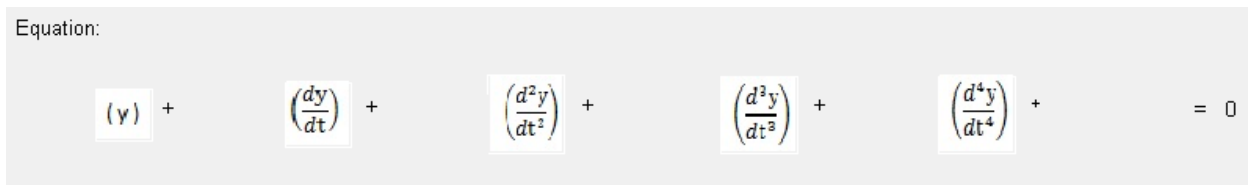


Figura 4.6.1 Ecuación por defecto

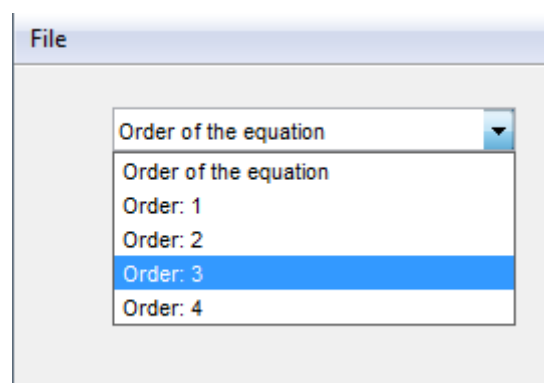


Figura 4.6.2 Selección grado de la ecuación

La primera opción es seleccionar el grado de la ecuación (orden de la derivada mayor); el orden máximo seleccionado es cuatro. Al hacerlo aparece, bajo el botón donde se ha seleccionado el orden de la ecuación, aparecen un número de cuadros ordenados igual al orden de la ecuación diferencial, figura 4.7.

Order: 1

Equation:

Coefficients and exponents:

First term

Coefficients: (y) Exp: ☐ delete term

Second term

$\left(\frac{dy}{dt}\right)$ ☐ delete term

INDEPENDENT TERM:

MAX. SIMULATION TIME (s):

Show equation (take care with parenthesis)

Figura 4.7 Selección de parámetros y coeficientes de la ecuación diferencial simple.

En primer cuadro ('Firstterm'), Figura 4.8, que corresponde al término en y (variable dependiente) de la ecuación, se introduce el coeficiente ('coefficients') del término y su exponente ('Exp') siempre entre paréntesis; en caso de no existir dicho término en la ecuación, se acciona el botón 'deleteterm' y el cuadro desaparece. Siguiendo el mismo procedimiento se introducen los términos (que existan) asociados a las sucesivas derivadas de y .

Coefficients and exponents:

First term

Coefficients: (y) Exp: ☐ delete term

Second term

$\left(\frac{dy}{dt}\right)$ ☐ delete term

Figura 4.8 Introducción del coeficiente y del exponente del término

Los coeficientes de los términos mencionados no son necesariamente constantes sino que pueden estar formados por funciones de las variables dependiente e independiente o pueden escribirse en forma paramétrica, es decir se introduce una cadena de letras que definen un parámetro al que más tarde se le asigna un valor. Para la correcta escritura de las funciones y parámetros dentro de los coeficientes se siguen reglas de escritura propias del lenguaje matemático: signos +, -, *, /, etc, y se definen las funciones igualmente. Con objeto de facilitar al usuario el trabajo, el botón 'Functions and symbols nomenclature' de la actual pantalla muestra el desplegable de ayuda de la figura 4.9 en el que se escriben las funciones más usadas, los símbolos de las operaciones matemáticas y algún dato más.

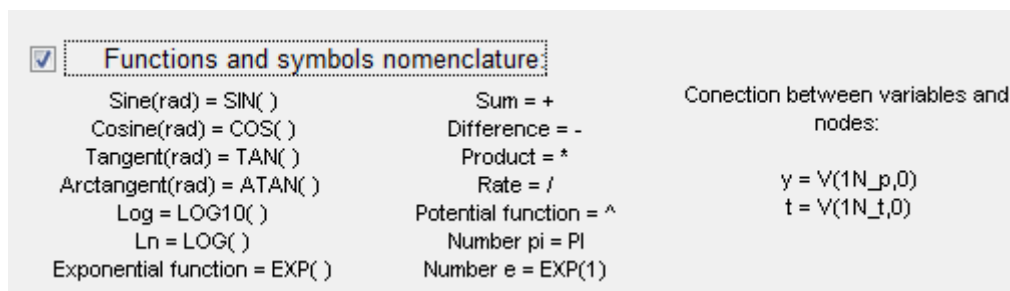


Figura 4.9 Desplegable del botón 'Functions and symbols nomenclature'

El término independiente (que puede ser también una función) se introduce en el cuadro 'Independentterm'. Para terminar se añade el tiempo máximo de simulación (el tiempo inicial siempre es cero) en el cuadro 'Max Simulation Time (s)'. Con todo, y tras revisar la escritura se acciona el botón 'Show equation' para que el programa nos muestre la ecuación del problema en la parte superior.

La Figura 4.10 es un ejemplo que corresponde a la selección de una ecuación de primer orden. En la propia pantalla se muestra la ecuación escrita con los signos

algebraicos de sus términos y los parámetros definidos en el problema (m_1 , k_1 y g en el término independiente y g en el término de la primera derivada).

Equation:

(1)

(2) $(y) + (3 \cdot g) \left(\frac{dy}{dt} \right) + (m_1 \cdot g \cdot k_1) = 0$

☐ delete term

INDEPENDENT TERM: $(m_1 \cdot g \cdot k_1)$

MAX. SIMULATION TIME (s): 20

Figura 4.10 Selección de una ecuación de primer orden.

La acción de pulsar el botón 'Show equation' tiene otro efecto, el de mostrar el recuadro 'Parameters and initialconditions' dentro de la misma pantalla a la derecha del botón anterior, Figura 4.10. Este nuevo botón da acceso a la pantalla 'Parameters and Initialconditions' que nos permite introducir tanto el valor de los parámetros como las condiciones iniciales del problema, figura 11. Se selecciona en primer lugar el número de parámetros del problema, que presenta bajo el botón 'Numbers of parameters' tantos recuadros como parámetros se hayan definido. Uno a uno se van introduciendo los parámetros con la sintaxis siguiente 'nombre del parámetro=valor'; por ejemplo $m_1=15$, $g=10$, $k_1=3$, etc.

Parameters and initial conditions

File

3

Other parameters:

Initial conditions (t=0):

Param 1: $m_1=15$

Param 2: $g=10$

Param 3: $k_1=3$

PRINT_STEP 1

RESULTS_DELAY 0

STEP_CEILING tmax / 50

RELTOL 1

$y = 1$

Figura 4.11 Pantalla de introducción de los parámetros y condiciones iniciales

Para completar la información del problema es necesario añadir otros parámetros relacionados con la simulación y los archivos tabulados de salida de datos. Estos son (figura 4.11): 'Printstep' que define el intervalo temporal con que se imprimen los resultados de la simulación (tablas de datos del archivo de extensión '.out' que elabora Pspice), 'Resultsdelay' que define el tiempo a partir del cual queremos imprimir los datos (generalmente cero), 'Stepceiling' que define el tiempo de paso cuando quiere ser impuesto por el usuario (ver manual del programa Pspice para mayor información sobre este parámetro), por defecto Pspice fija un valor igual al tiempo máximo de simulación dividido por 50, y 'Reltol' parámetro asociado a la precisión impuesta a los cálculos numéricos. Un valor mal elegido de este parámetro en seguida produce falta de convergencia por lo que es preciso establecerlo adecuadamente (ver manual de Pspice) el rango de este parámetro.

Para terminar, se introducen en la misma pantalla los valores iniciales del problema: valor de la variable dependiente y sus derivadas en el instante inicial. Para una ecuación de segundo orden se precisa el valor de la función y su primera derivada; para una ecuación de primer orden sólo el de la función, figura 4.11. El botón 'Save' en la parte inferior de la pantalla guarda todos los datos introducidos. La opción 'File' → 'Clear', dentro de esta pantalla, borra todos los datos para introducir datos nuevos, figura 4.12.

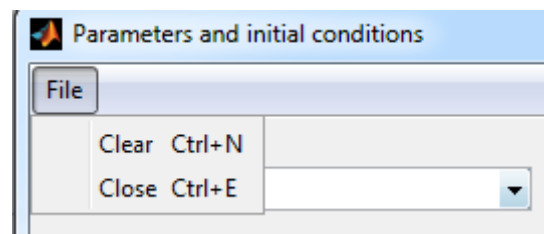


Figura 4.12 Ruta para reescribir los parámetros y condiciones iniciales

Finalizada la entrada de datos que definen el problema y pulsando 'Save', se cierra esta pantalla y el programa vuelve a la pantalla anterior en la que aparecen nuevas opciones, botones 'Createtext file', 'Simulation' y 'Cancel', Figura 4.13.

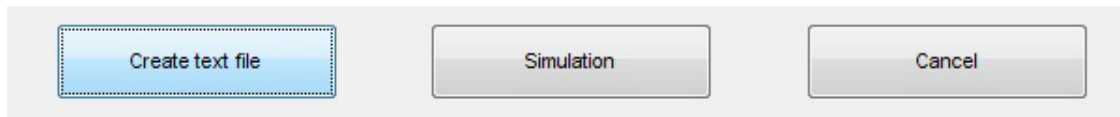


Figura 4.13 Opciones tras la introducción de datos

El botón 'Createtext file' genera el archivo de texto del modelo y lo presenta en el block de notas (Note Pad) del ordenador para poder ser revisado y/o modificado por el usuario. Si se desean ejecutar las modificaciones hechas por el usuario en el block de notas es preciso convertir el archivo en uno de extensión '.cir' y ejecutarlo separadamente en Pspice ya que dichas modificaciones no alteran el archivo de texto creado por CODENS_14. El objeto de presentar el archivo de texto del modelo al usuario es tan solo para inspección y comprobación del mismo. Así, la opción 'Simulación' arranca Pspice y ejecuta el modelo creado por CODENS_14.

En todo momento, como novedad en CODENS_14, si algún valor es introducido de manera errónea o falta algún valor, el programa nos avisará con un mensaje de error, figura 4.14. En él aparecerá el mensaje "Falta algún valor" y "Algún valor es incorrecto". Esto nos ayudará a evitar posibles fallos posteriores debido al despiste o datos erróneos, ya que no nos permitirá avanzar hasta solucionar el fallo.

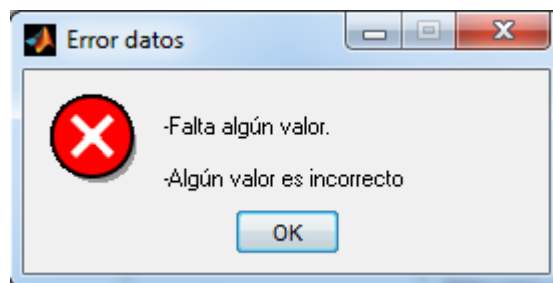


Figura 4.14 Mensaje de error en los valores introducidos.

En el botón 'File' de la pantalla actual 'Ordinarydifferentialequation' figura 4.15, podemos encontrar las opciones: 'New' para diseñar el modelo de un nuevo problema, 'Open' para llamar a un modelo ya creado y guardado, 'Save' activada ahora para guardar el modelo que acaba de ser diseñado, y 'Close' para cerrar la pantalla. Todos los modelos generados por CODENS_14 se guardan en una carpeta de nombre 'Saveexercise' instalada en la misma ruta del programa.

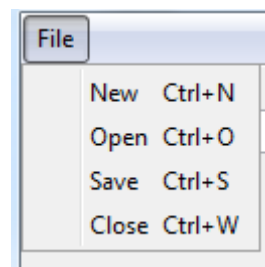


Figura 4.15 Opciones de salvar, guardar y abrir archivos

La simulación del modelo (opción 'Simulation' de la Figura 4.13) abre Pspice y ejecuta el programa. No nos extendemos aquí en explicaciones sobre los entornos de simulación y gráfico de Pspice aunque los presentamos en las >Figuras 4.16 y 4.17, pues no es objeto de esta memoria. Remitimos para ello al manual correspondiente.

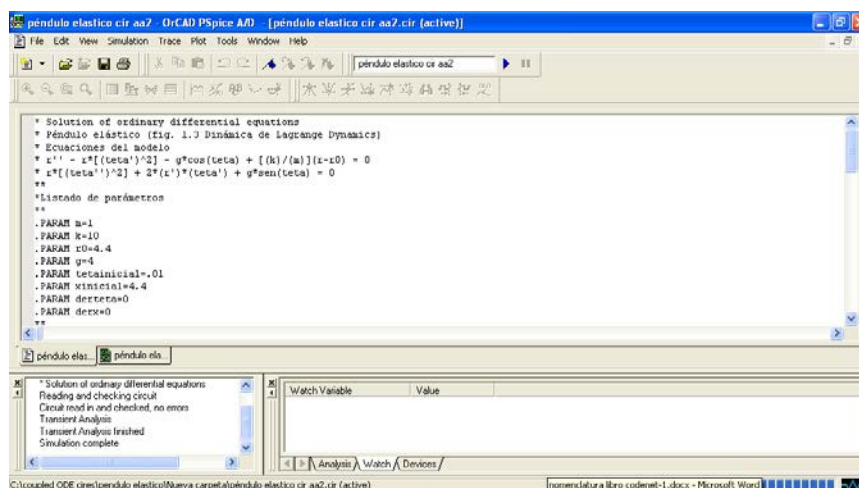


Figura 4.16 Entorno de simulación de Pspice

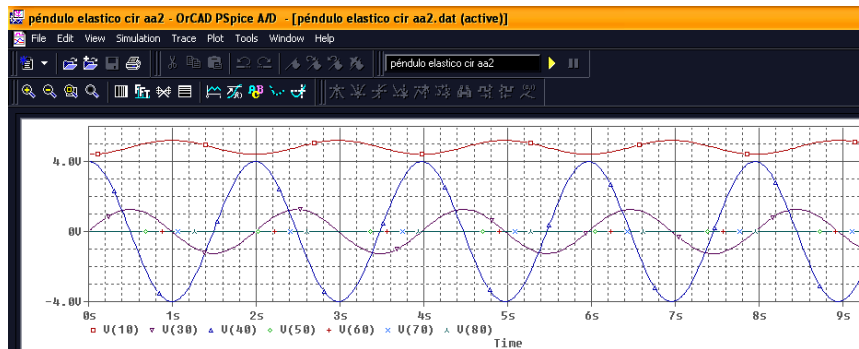


Figura 4.17 Entorno gráfico de salida de Pspice

Una vez finalizada la simulación en Pspice, aunque queda abierto en entorno de este programa aparece la pantalla mostrada en la Figura 4.18 para que el usuario seleccione las gráficas que desea ver en Matlab. CODENS_14 importa los resultados tabulados de la simulación desde Pspice a Matlab mediante rutinas específicas y genera los gráficos. Las opciones de representación son: 'show y(t)' que muestra la solución $y=y(t)$, 'show $dy(t)/dt$ ' que representa la primera derivada en función del tiempo, 'show $d^2y(t)/dt^2$ ' que muestra la segunda derivada y 'show $y=f(dy(t)/dt)$ ' que muestra el diagrama de fases o representación de y en función de su primera derivada dy/dt . Por último, la opción 'show all graphs' muestra simultáneamente todos los gráficos anteriores.

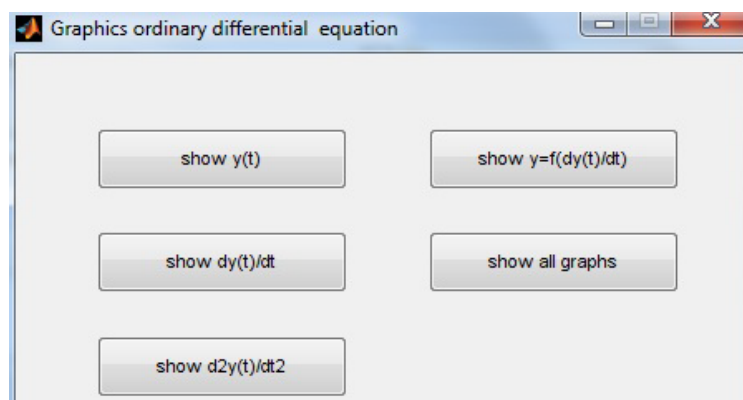


Figura 4.18 Opciones de representación gráfica

Ejercicios con ecuaciones diferenciales acopladas

Lo primero es definir el número de ecuaciones del sistema (que coincide en número de variables). Tras seleccionar el botón 'System of coupled, ordinarydiferentialequations' de la pantalla de la figura 4.2, el programa da acceso a la pantalla mostrada en la figura 4.19. Una vez seleccionado el número de ecuaciones la pantalla se modifica introduciendo nuevos botones para acceder a la definición de las ecuaciones y parámetros, Figura 4.20.

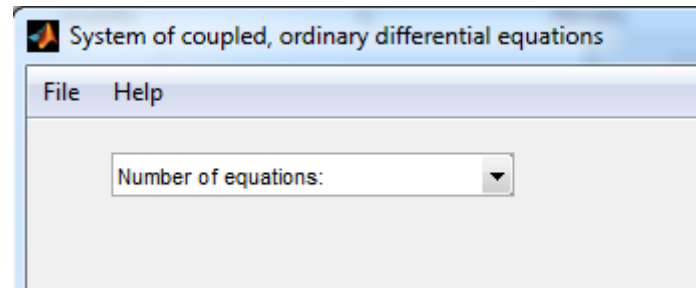


Figura 4.19 Selección del número de ecuaciones en la opción 'System of coupled, ordinarydiferentialequations'

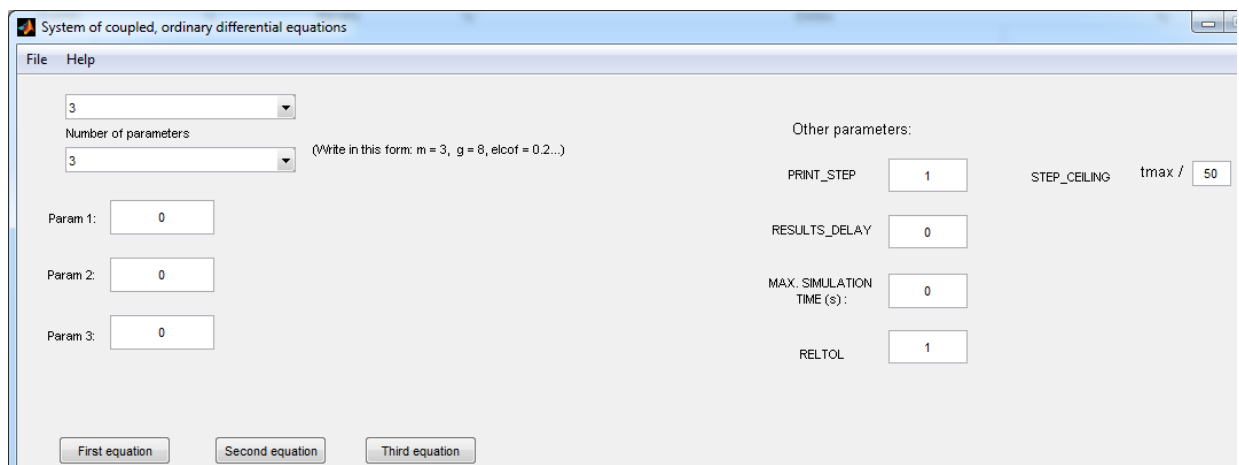


Figura 4.20 Pantalla de selección de las ecuaciones y los parámetros del problema en la opción 'System of coupled, ordinarydiferentialequations'

En relación con la introducción de los parámetros del problema, se abre una pestaña bajo el recuadro del número de ecuaciones en la que se selecciona el número de parámetros, figura 4.21.

3

Number of parameters

10

(Write in this form: m = 3, g = 8, elcof = 0.2...)

Number of parameters:

1

2

3

4

5

6

7

8

9

10

Param 1: 0

Param 2: 0

Param 3: 0

Param 4: 0

Param 5: 0

Param 6: 0

Param 7: 0

Param 8: 0

Param 9: 0

Param 10: 0

First equation

Second equation

Third equation

Figura 4.21 Selección número de parámetros.

Una vez definido este número se abren tantos recuadros como número de parámetros tenga el problema. En ellos se introducen los valores de los parámetros con la sintaxis indicada en el caso de problemas de una sola ecuación ('nombre del parámetro'='valor del parámetro'). Los parámetros se definen con caracteres alfanuméricos; ejemplo: m1=0.15, alfa = 10.45, gravedad = 10. La introducción de los parámetros se realiza de igual manera que en la opción de ecuaciones diferenciales simples explicada en el apartado anterior. Como novedad en CODENS_14, al aumentar o disminuir el número de parámetros, no se borrarán los valores introducidos como pasaba en la versión anterior, sino que se mantendrán los parámetros deseados para evitar introducirlos de nuevo.

Introducidos y salvados todos los parámetros, incluidos los relacionados con la simulación ('Prinpsstep', Resultsdelay', 'Max simulation time (s)' y 'Rentol'), el programa da acceso a las pantallas que definen cada una de las ecuaciones del sistema (pantallas similares al caso de problemas no acoplados), Figura 4.22. Para ello se accionan sucesivamente los recuadros 'Firstequation', 'Secondequation', 'Thirdequation' y 'Fourthequation' de la pantalla de la Figura 4.19. Para cada una de las ecuaciones se

selecciona, en primer lugar, su orden, y a continuación se introducen los coeficientes y exponentes de sus términos.

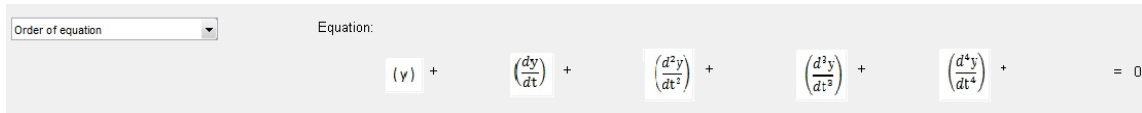


Figura 4.22 Pantalla de definición del orden de cada ecuación

La introducción de los coeficientes y exponentes de cada una de las ecuaciones del sistema se introduce con la misma secuencia y reglas que en el caso de problemas de ecuaciones no acopladas, Figura 4.23. Ahora cada variable se le añade un número asociado a su ecuación correspondiente: La primera ecuación corresponderá con la variable y_1 , la segunda con y_2 y así sucesivamente con el número de ecuaciones que deseemos introducir. Los valores introducidos deben ir entre paréntesis para evitar posibles errores con el programa.

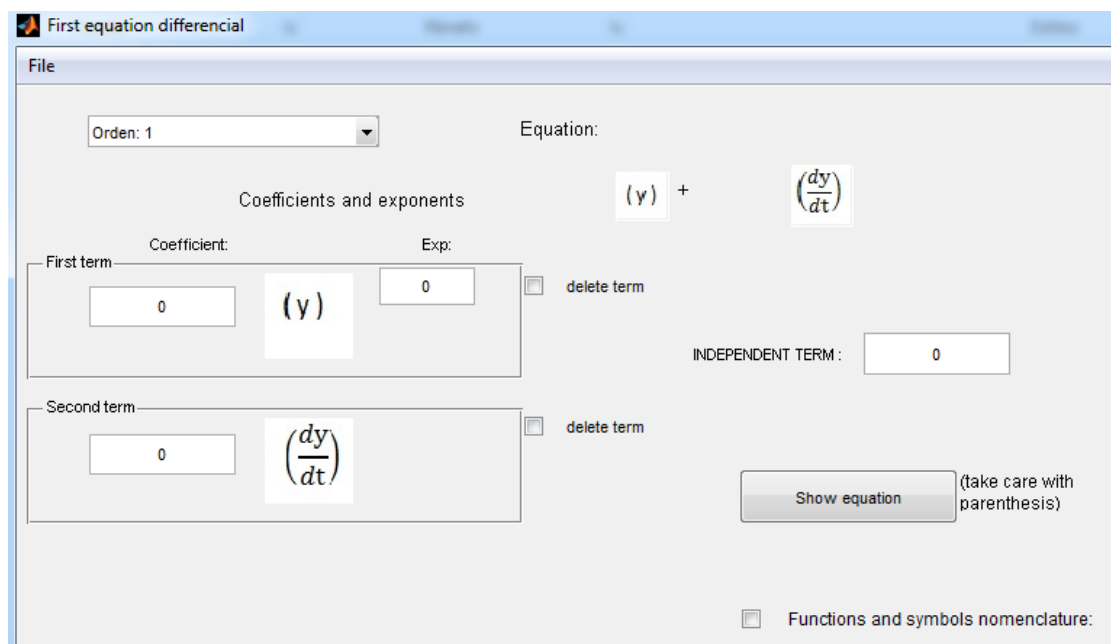


Figura 4.23 Pantalla de introducción de parámetros, exponentes y términos independientes

Para facilitar una sintaxis correcta en la expresión de los coeficientes, cuando estos contienen cualquier variable dependiente o cualquier derivada de estas variables, se

incorpora un listado con la escritura de estas funciones que se muestra accionando el botón 'coefficients and exponentslist'. Este botón da acceso, en primer lugar, a la posición en la que se desea introducir la variable (o su derivada de cualquier orden), Figura 4.24 y, en segundo lugar, a la elección de la variable, Figura 4.25. El significado de la posición es intuitivo; 'coefficientfirstterm' alude al coeficiente del primer término (o término de la variable y , Figura 4.24), con lo que en ese lugar se escribirá la variable seleccionada; 'coefficientsecondterm' alude al coeficiente del segundo término (o término de la derivada dy/dt)..., y lo mismo con 'exponentfirstterm' y 'exponentsecondterm' que aluden a los exponentes del primer y segundo términos, respectivamente. Y así con el resto de las posiciones. 'Independentterm' por último indica la escritura de la variable correspondiente en el término independiente.

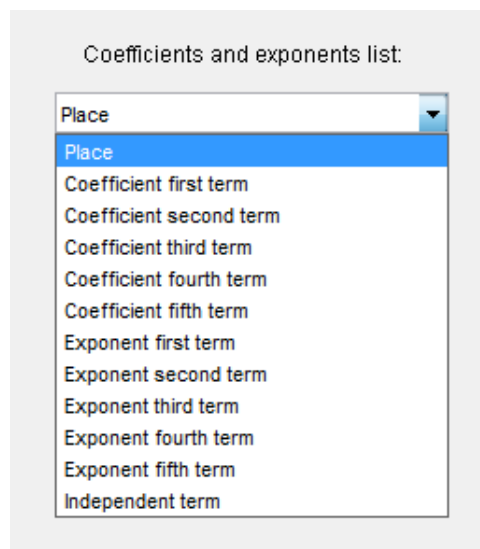


Figura 4.24 Desplegable del lugar en donde se desea escribir la variable.

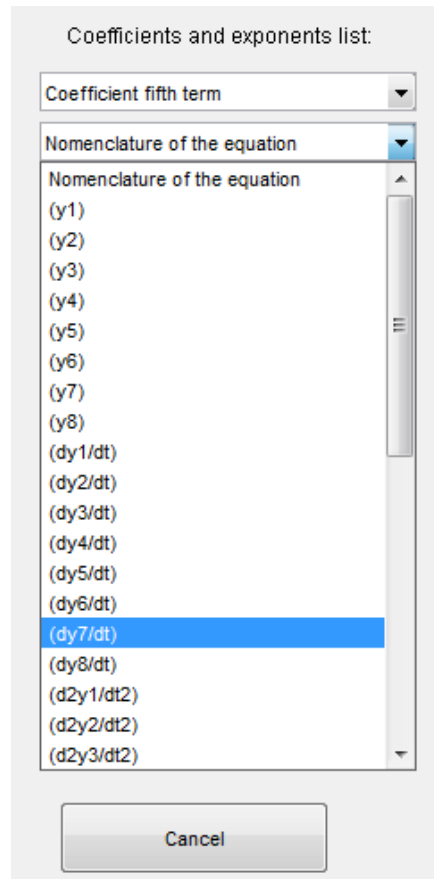


Figura 4.25 Desplegable de la selección de la variable o cualquiera de sus derivadas

Cada vez que se selecciona una variable o cualquiera de sus derivadas es preciso pulsar 'Introduce' para que CODENS_14 las escriba en el lugar previsto. El resto de los comandos de la pantalla de la Figura 4.23 ó 4.24 tienen el mismo significado que el explicado en el caso de ecuaciones no acopladas.

Si algún valor es introducido de manera errónea o falta algún valor, el programa nos avisará con un mensaje de error, figura 4.26. En él aparecerá el mensaje "Falta algún valor" y "Algún valor es incorrecto". Esto nos ayudará a evitar posibles fallos posteriores debido al despiste o datos erróneos, ya que no nos permitirá avanzar hasta solucionar el fallo.

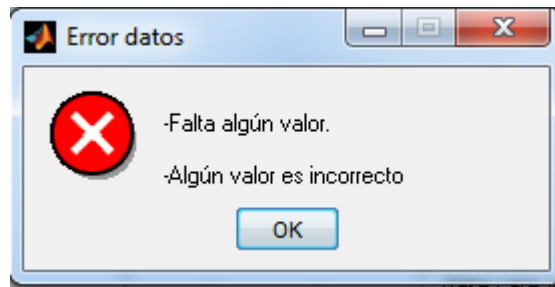


Figura 4.26 Mensaje de error en los valores introducidos.

Para cada ecuación, una vez introducidos los datos de la misma, se accede a su representación escrita mediante el botón 'Show equation'; un ejemplo se muestra en la Figura 4.27. Tras comprobar que es correcta es preciso guardarla antes de introducir la siguiente ecuación. Para finalizar, deben introducirse las condiciones iniciales de las variables y sus correspondientes necesarias derivadas, si las hubiere. Figura 4.28.

Orden: 1

Equation: (3)

Coefficients and exponents

First term: Coefficient: (2) Exp: (3) delete term

Second term: Coefficient: (m1) Exp: $\left(\frac{dy}{dt}\right)$ delete term

INDEPENDENT TERM: (g*k1)

Equation: $(2) (y) + (m1) \left(\frac{dy}{dt}\right) + (g*k1) = 0$

show equation (take care with parenthesis)

Functions and symbols nomenclature

Sine(rad) = SIN()	Sum = +	Connection between variables and nodes: y(1) = V(1N_p_y1,0) y(2) = V(1N_p_y2,0) y(3) = V(1N_p_y3,0) ... t = V(1N_t,0)
Cosine(rad) = COS()	Difference = -	
Tangent(rad) = TAN()	Product = *	
Arctangent(rad) = ATAN()	Rate = /	
Log = LOG10()	Potential function = ^	
Ln = LOG()	Number pi = PI	
Exponential function = EXP()	Number e = EXP(1)	
	Raiz cuadrada = SQRT()	

Figura 4.27 Representación de la ecuación una vez introducidos sus correspondientes coeficientes y exponentes

First equation ☒

y1 = 0

Figura 4.28 Sección de introducción de condiciones iniciales

Una vez introducidas todas las ecuaciones se procede de igual forma al caso de ecuaciones no acopladas; el modelo puede salvarse, botón 'Save' de la Figura 4.13 (para ecuaciones acopladas) y/o simularse, botón 'Simulation'. En el primer caso el archivo de texto se presenta en la pantalla en el block de notas para ser visto por el usuario y se guarda en la carpeta 'Saveexersice'. Es también aplicable lo dicho anteriormente en relación con las posibilidades de salvar, guardar o abrir archivos, figura 4.29.

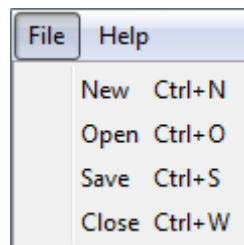


Figura 4.29 Opciones de manipulación de archivos

Una vez ejecutado el programa mediante el botón 'simulation' de la Figura 4.13, se abre el entorno de Pspice y se muestra la pantalla de la Figura 4.30, 'Show simulationresults' para que el usuario seleccione las gráficas que desea ver en Matlab. Para ello, CODENS_14 importa los resultados tabulados de la simulación desde Pspice a Matlab y genera los gráficos solicitados. Las opciones de representación permiten representar todas las funciones del problema, sus correspondientes derivadas de cualquier orden y cualquier tipo de diagrama que represente una función frente cualquiera de sus derivadas o cualquiera de las derivadas frente a otras derivadas de todas las funciones.

En el cuadro 'Bottongroup' de la Figura se insertan las variables de los ejes de abscisas y ordenadas, seleccionándolas de los cuadros 'Firstequation', 'Secondequation', 'thirdequation'... y el tiempo; en cada uno de estos cuadros aparece el listado completo de variables del problema y sus derivadas. El proceso es seleccionar la magnitud a representar y pulsar el eje en el que va insertada en el

‘Bottongroup’, en el que aparece la variable seleccionada como indica la Figura 4.31. En el capítulo de aplicaciones se muestra la forma de estos gráficos.

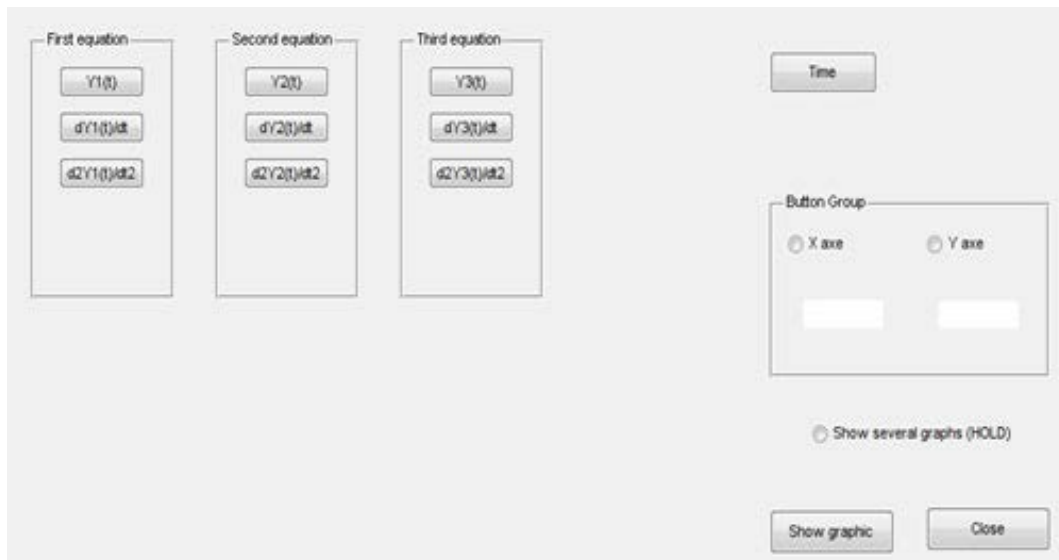


Figura 4.30 Opciones de representación gráfica para los problemas de ecuaciones acopladas

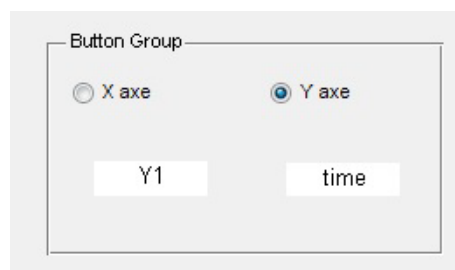


Figura 4.31 Variables a representar

4.2.2 Nomenclatura de componentes y nodos

Para facilitar la asociación de elementos y nudos de los circuitos principales y auxiliares con las variables y sus sucesivas derivadas se ha establecido una nomenclatura coherente (aunque puede resultar en una cadena larga en ocasiones) que describimos a continuación.

Caso de problemas formulados por una sola ecuación

1. Nomenclatura de Fuentes

G → Fuente de corriente controlada por tensión

E → Fuente de tensión controlada por tensión

H → Fuente de tensión controlada por corriente

F → Fuente de corriente controlada por corriente

1.1) Fuentes del circuito principal

G_N → fuente G asociada a la derivada $d^n y/dt^n$

G_3 → fuente G asociada a la derivada $d^3 y/dt^3$

G_2 → fuente G asociada a la derivada $d^2 y/dt^2$

G_1 → fuente G asociada a la derivada dy/dt

G_0 → fuente G asociada al término y

G_i → fuente G asociada al término independiente

1.2) Fuentes de los circuitos auxiliares para generar las sucesivas derivadas

E_1 → fuente E asociada a la generación de la tensión $y(t)$

(este circuito auxiliar contiene un condensador, una pila medidora de corriente y una resistencia infinita a la salida de la fuente)

H_1 → fuente H asociada a la generación de la tensión dy/dt

(este circuito auxiliar contiene un condensador, una pila medidora de corriente y una resistencia infinita a la salida de la fuente)

H_2 → fuente H asociada a la generación de la tensión $d^2 y/dt^2$

H_N → fuente H asociada a la generación de la tensión $d^n y/dt^n$

(este circuito auxiliar contiene una resistencia de valor unidad en paralelo con la fuente)

1.3) Fuentes del circuito auxiliar para generar la variable tiempo

V_time → fuente E asociada a la generación de la tensión $y(t)$

(es una fuente lineal a tramos, de un solo tramo de pendiente 45º, con una resistencia unidad en su salida)

2. Nomenclatura de resistencias

R_{inf_p} → Resistencia de valor infinito del circuito principal

$R_{inf_a_E1}$ → Resistencia de valor infinito del circuito auxiliar asociado a E_1

$R_{inf_a_H1}$ → Resistencia de valor infinito del circuito auxiliar asociado a H_1

$R_{inf_a_H2}$ → Resistencia de valor infinito del circuito auxiliar asociado a H_2

$R_{uni_a_HN}$ → Resistencia de valor unidad del circuito auxiliar asociado a H_N
(en sus extremos se obtiene el valor $d^n y/dt^n$)

R_{time} → Resistencia de valor unidad del circuito auxiliar generador V_{time} de la variable tiempo (en sus extremos se obtiene la variable tiempo)

3. Nomenclatura de condensadores

C_{E1} → Condensador del circuito auxiliar E_1 (su corriente es la derivada dy/dt)

C_{H1} → Condensador del circuito auxiliar H_1 (su corriente es la derivada dy^2/dt^2)

C_{H2} → Condensador del circuito auxiliar H_2 (su corriente es la derivada dy^3/dt^3)

... (C_{HN} no existe)

4. Nomenclatura de baterías nulas (que actúan como medidores de intensidad)

V_{nula_E1} → Batería nula asociada al circuito auxiliar E_1 (su corriente es la derivada dy/dt)

V_{nula_H1} → Batería nula asociada al circuito auxiliar H_1 (su corriente es la derivada $d^2 y/dt^2$)

V_{nula_H2} → Batería nula asociada al circuito auxiliar H_2 (su corriente es la derivada $d^3 y/dt^3$)

... (V_{nula_HN} no existe)

5. Nomenclatura de nodos

5.1) Circuito principal

En el circuito principal sólo está el nodo principal, cuyo voltaje se ajusta a la solución $y(t)$ del problema, más el nodo de masa (común a este circuito y al resto de los circuitos auxiliares).

N_{p_y} → Nodo del circuito principal de la variable y_1 . En este nodo está la solución $y_1(t)$

0 → Nodo común (masa)

N_{time} → Nodo del circuito auxiliar de la variable tiempo. En este nodo está el valor instantáneo de t .

5.2) Circuitos auxiliares

En cada circuito auxiliar hay un nodo que se define asociándole la correspondiente fuente

N_{E1} → Nodo del circuito auxiliar E_1 . En este nodo está la solución $y(t)$

N_{H1} → Nodo del circuito auxiliar H_1 . En este nodo está la solución dy/dt

N_{H2} → Nodo del circuito auxiliar H_1 . En este nodo está la solución d^2y/dt^2

...

N_{HN} → Nodo del circuito auxiliar H_N . En este nodo está la solución d^ny/dt^n

Nodos intermedios:

En los circuitos auxiliares existen nudos intermedios únicamente entre los condensadores y las baterías nulas. Estos nudos tienen nombres con raíz ' $N_{aux_}$ ' y se asocian a las fuentes de cada circuito. Así, estos nudos quedan de la siguiente forma:

N_{aux_E1}

N_{aux_H1}

N_{aux_H2}

...

N_{aux_HN}

5.3) Nudos (tensiones) de localización de la variable tiempo y de las soluciones

$t(\text{tiempo})$ \longleftrightarrow N_t

$y(t)$	\longleftrightarrow	N_p	\longleftrightarrow	N_{E1}
dy/dt	\longleftrightarrow	N_{H1}		
d^2y/dt^2	\longleftrightarrow	N_{H2}		
d^3y/dt^3	\longleftrightarrow	N_{H3}		

Caso de problemas formulados mediante sistemas de ecuaciones

6. Nomenclatura de Fuentes

6.1) Fuentes del circuito principal de la ecuación de y_1

$G_N_y1 \rightarrow$ fuente G asociada a la derivada $d^n y_1/dt^n$

...

$G_3_y1 \rightarrow$ fuente G asociada a la derivada $d^3 y_1/dt^3$

$G_2_y1 \rightarrow$ fuente G asociada a la derivada $d^2 y_1/dt^2$

$G_1_y1 \rightarrow$ fuente G asociada a la derivada dy_1/dt

$G_0_y1 \rightarrow$ fuente G asociada al término y_1

$G_i_y1 \rightarrow$ fuente G asociada al término independiente de la ecuación de y_1

Si existen varias fuentes independientes podrían definirse en la forma G_i1_y1 , G_i2_y1 ..., aunque para ello se debería trabajar sobre el archivo de texto ya que CODENS_13 sólo implementa una fuente por cada ecuación, cualquiera que sea la forma del término independiente de ésta.

Fuentes del circuito principal de la ecuación de y_2

$G_N_y2 \rightarrow$ fuente G asociada a la derivada $d^n y_2/dt^n$

...

$G_3_y2 \rightarrow$ fuente G asociada a la derivada $d^3 y_2/dt^3$

$G_2_y2 \rightarrow$ fuente G asociada a la derivada $d^2 y_2/dt^2$

$G_1_y2 \rightarrow$ fuente G asociada a la derivada dy_2/dt

$G_0_y2 \rightarrow$ fuente G asociada al término y_2

$G_i_y2 \rightarrow$ fuente G asociada al término independiente de la ecuación de y_2

Aplica lo dicho anteriormente en relación con las fuentes asociadas a términos independientes.

Fuentes del circuito principal de la ecuación de y_3

$G_N_y3 \rightarrow$ fuente G asociada a la derivada $d^n y_3 / dt^n$

...

$G_3_y3 \rightarrow$ fuente G asociada a la derivada $d^3 y_3 / dt^3$

$G_2_y3 \rightarrow$ fuente G asociada a la derivada $d^2 y_3 / dt^2$

$G_1_y3 \rightarrow$ fuente G asociada a la derivada dy_3 / dt

$G_0_y3 \rightarrow$ fuente G asociada al término y_3

$G_i_y3 \rightarrow$ fuente G asociada al término independiente de la ecuación de y_3
(ídem lo mencionado anteriormente)

6.2) Fuentes de los circuitos auxiliares para generar las sucesivas derivadas de cada variable

6.2.1) Fuentes de los circuitos auxiliares de la variable y_1

$E_1_y1 \rightarrow$ fuente E asociada a la generación de la tensión $y_1(t)$

(este circuito auxiliar contiene un condensador y una pila medidora de corriente)

$H_1_y1 \rightarrow$ fuente H asociada a la generación de la tensión dy_1 / dt (ídem)

$H_2_y1 \rightarrow$ fuente H asociada a la generación de la tensión $d^2 y_1 / dt^2$ (ídem)

$H_3_y1 \rightarrow$ fuente H asociada a la generación de la tensión $d^3 y_1 / dt^3$ (ídem)

...

$H_N_y1 \rightarrow$ fuente H asociada a la generación de la tensión $d^n y_1 / dt^n$

(este circuito auxiliar contiene una resistencia de valor unidad)

i.2.2) Fuentes de los circuitos auxiliares de la variable y_2

$E_1_y2 \rightarrow$ fuente E asociada a la generación de la tensión $y_2(t)$

(este circuito auxiliar contiene un condensador y una pila medidora de corriente)

$H_1_y2 \rightarrow$ fuente H asociada a la generación de la tensión dy_2 / dt (ídem)

$H_2_y2 \rightarrow$ fuente H asociada a la generación de la tensión $d^2 y_2 / dt^2$ (ídem)

$H_3_y2 \rightarrow$ fuente H asociada a la generación de la tensión $d^3 y_2 / dt^3$ (ídem)

...

$H_N_y2 \rightarrow$ fuente H asociada a la generación de la tensión $d^n y_2 / dt^n$
 (este circuito auxiliar contiene una resistencia de valor unidad)

i.2.3) Fuentes de los circuitos auxiliares de la variable y_3

...

6.3) Fuente del circuito auxiliar para generar la variable tiempo

$V_time \rightarrow$ fuente E asociada a la generación de la tensión $y(t)$
 (es una fuente lineal a tramos, de un solo tramo de pendiente 45º,
 con una
 resistencia unidad en su salida)

7. Nomenclatura de resistencias

Resistencias asociadas a la variable y_1

$R_inf_p_y1 \rightarrow$ Resistencia de valor infinito del circuito principal de la variable y_1

$R_inf_p_y2 \rightarrow$ Resistencia de valor infinito del circuito principal de la variable y_2

$R_inf_p_y3 \rightarrow$ Resistencia de valor infinito del circuito principal de la variable y_3

...

$R_inf_a_E1_y1 \rightarrow$ Resistencia de valor infinito del circuito auxiliar asociado a
 E_1_y1

$R_inf_a_H1_y1 \rightarrow$ Resistencia de valor infinito del circuito auxiliar asociado a
 H_1_y1

$R_inf_a_H2_y1 \rightarrow$ Resistencia de valor infinito del circuito auxiliar asociado a
 H_2_y1

...

$R_1_a_HN_y1 \rightarrow$ Resistencia de valor unidad del circuito auxiliar asociado a
 H_N_y1

(en sus extremos está la tensión $d^n y_1 / dy_1^n$)

Resistencias asociadas a la variable y_2

$R_inf_p_y2 \rightarrow$ Resistencia de valor infinito del circuito principal

$R_inf_a_E1_y2 \rightarrow$ Resistencia de valor infinito del circuito auxiliar asociado a
 E_1_y2

$R_{inf_a_H1_y2}$ → Resistencia de valor infinito del circuito auxiliar asociado a H_{1_y2}

$R_{inf_a_H2_y2}$ → Resistencia de valor infinito del circuito auxiliar asociado a H_{2_y2}

...

$R_{1_a_HN_y2}$ → Resistencia de valor unidad del circuito auxiliar asociado a H_{N_y2}

(en sus extremos está la tensión $d^n y_2 / dy_2^n$)

Resistencias asociadas a la variable y_3

...

Resistencia asociadas a la generación de la variable tiempo

R_{time} → Resistencia de valor unidad del circuito auxiliar del generador V_{time} asociado a la variable tiempo

(la tensión en sus extremos es la variable independiente tiempo)

8. Nomenclatura de condensadores

C_{E1_y1} → Condensador del circuito auxiliar E_{1_y1} (su corriente es la derivada dy_1/dt)

C_{H1_y1} → Condensador del circuito auxiliar H_{1_y1} (su corriente es la derivada dy_1^2/dt^2)

C_{H2_y1} → Condensador del circuito auxiliar H_{2_y1} (su corriente es la derivada dy_1^3/dt^3)

...

(C_{HN-y1} no existe)

C_{E1_y2} → Condensador del circuito auxiliar E_{1_y2} (su corriente es la derivada dy_2/dt)

C_{H1_y2} → Condensador del circuito auxiliar H_{1_y2} (su corriente es la derivada dy_2^2/dt^2)

C_{H2_y2} → Condensador del circuito auxiliar H_{2_y2} (su corriente es la derivada dy_2^3/dt^3)

...

(C_{HN-y1} no existe)

$C_{E1_y3} \rightarrow$ Condensador del circuito auxiliar E_{1_y3} (su corriente es la derivada dy_3/dt)

$C_{H1_y3} \rightarrow$ Condensador del circuito auxiliar H_{1_y3} (su corriente es la derivada dy_3^2/dt^2)

$C_{H2_y3} \rightarrow$ Condensador del circuito auxiliar H_{2_y3} (su corriente es la derivada dy_3^3/dt^3)

... (C_{HN-y1} no existe)

9. Nomenclatura de baterías nulas (que actúan como medidores de intensidad)

$V_{nula_E1_y1} \rightarrow$ Batería nula asociada al circuito auxiliar E_{1_y1} (su corriente es la derivada dy_1/dt)

$V_{nula_H1_y1} \rightarrow$ Batería nula asociada al circuito auxiliar H_{1_y1} (su corriente es la derivada d^2y_1/dt^2)

$V_{nula_H2_y1} \rightarrow$ Batería nula asociada al circuito auxiliar H_{2_y1} (su corriente es la derivada d^3y_1/dt^3)

...

($V_{nula_HN_y1}$ no existe)

$V_{nula_E1_y2} \rightarrow$ Batería nula asociada al circuito auxiliar E_{1_y2} (su corriente es la derivada dy_2/dt)

$V_{nula_H1_y2} \rightarrow$ Batería nula asociada al circuito auxiliar H_{1_y2} (su corriente es la derivada d^2y_2/dt^2)

$V_{nula_H2_y2} \rightarrow$ Batería nula asociada al circuito auxiliar H_{2_y2} (su corriente es la derivada d^3y_2/dt^3)

...

($V_{nula_HN_y2}$ no existe)

...

10. Nomenclatura de nodos

10.1) Circuito principal

En el circuito principal i de la ecuación asociada a la variable y_i hay dos nodos, el principal (cuya tensión es la solución $y_i(t)$), más el nodo de masa (común a todos los circuitos principales y auxiliares). Así, existen tantos nodos principales como variables o como circuitos principales.

N_p_y1 → Nodo del circuito principal del circuito principal de y_1 . En este nodo está la solución $y_1(t)$

N_p_y2 → Nodo del circuito principal del circuito principal de y_2 . En este nodo está la solución $y_2(t)$

N_p_y3 → Nodo del circuito principal del circuito principal de y_3 . En este nodo está la solución $y_3(t)$

0 → Nodo común (masa)

10.2) Circuitos auxiliares

10.2.1) Nodos principales de los circuitos auxiliares:

Por cada variable, en cada uno de los circuitos auxiliares vinculados a ésta, hay un nodo que se define asociándole la correspondiente fuente

N_E1_y1 → Nodo del circuito auxiliar E_1_y1 . En este nodo está la solución $y_1(t)$

N_H1_y1 → Nodo del circuito auxiliar H_1_y1 . En este nodo está la solución dy_1/dt

N_H2_y1 → Nodo del circuito auxiliar H_1_y1 . En este nodo está la solución d^2y_1/dt^2

...

N_HN_y1 → Nodo del circuito auxiliar H_N_y1 . En este nodo está la solución d^ny_1/dt^n

N_E1_y2 → Nodo del circuito auxiliar E_1_y2 . En este nodo está la solución $y_2(t)$

N_H1_y2 → Nodo del circuito auxiliar H_1_y2 . En este nodo está la solución dy_2/dt

N_H2_y2 → Nodo del circuito auxiliar H_1_y2 . En este nodo está la solución d^2y_2/dt^2

...

N_HN_y2 → Nodo del circuito auxiliar H_N_y2 . En este nodo está la solución d^ny_2/dt^n

N_E1_y3 → Nodo del circuito auxiliar E_1_y3 . En este nodo está la solución $y_3(t)$

N_H1_y3 → Nodo del circuito auxiliar H_1_y3 . En este nodo está la solución dy_3/dt

N_H2_y3 → Nodo del circuito auxiliar H_1_y3 . En este nodo está la solución d^2y_3/dt^2

...

N_HN_y3 → Nodo del circuito auxiliar H_N_y3 . En este nodo está la solución d^ny_3/dt^n

10.2.2) Nodos intermedios de los circuitos auxiliares

En los circuitos auxiliares existen nodos intermedios únicamente entre los condensadores y las baterías nulas. Estos nodos tienen nombres con raíz 'N_aux_' y se asocian a las fuentes de cada circuito. Así, quedan de la siguiente forma:

Nodos intermedios de los circuitos auxiliares asociados a la variable y_1 :

N_aux_E1_y1
 N_aux_H1_y1
 N_aux_H2_y1
 ...
 N_aux_HN_y1

Nodos intermedios de los circuitos auxiliares asociados a la variable y_2 :

N_aux_E1_y2
 N_aux_H1_y2
 N_aux_H2_y2
 ...
 N_aux_HN_y2
 ...

11. Nodos (tensiones) de localización de las soluciones y de la variable tiempo

$t(\text{tiempo})$	\longleftrightarrow	N_t	
$y_1(t)$	\longleftrightarrow	N_p_y1	\longleftrightarrow N_E1_y1)
dy_1/dt	\longleftrightarrow	N_H1_y1	
d^2y_1/dt^2	\longleftrightarrow	N_H2_y1	
d^3y_1/dt^3	\longleftrightarrow	N_H3_y1	
...			
$y_2(t)$	\longleftrightarrow	N_p_y2	\longleftrightarrow N_E1_y2)
dy_2/dt	\longleftrightarrow	N_H1_y2	
d^2y_2/dt^2	\longleftrightarrow	N_H2_y2	
d^3y_2/dt^3	\longleftrightarrow	N_H3_y2	
...			
$y_3(t)$	\longleftrightarrow	N_p_y3	\longleftrightarrow N_E1_y3)

$$\begin{array}{ll}
 dy_3/dt & \longleftrightarrow N_H1_y3 \\
 d^2y_3/dt^2 & \longleftrightarrow N_H2_y3 \\
 d^3y_3/dt^3 & \longleftrightarrow N_H3_y3 \\
 \dots &
 \end{array}$$

4.3 Estructura de los archivos de texto

Una vez diferenciados los circuitos principales y auxiliares del modelo y establecida la nomenclatura de elementos y nodos de ambos circuitos, es preciso organizar la estructura del archivo de texto del modelo en red con objeto de que el usuario reconozca sus diferentes partes y tenga acceso a un rápido reconocimiento de sus partes y consiguiente manipulación o corrección si procede. Debe recordarse que la manipulación del modelo, una vez guardado, no cambia el propio modelo y ha guardado. Si queremos incluir rectificaciones hay que guardarlo con la extensión '.cir' y asignándole un nuevo nombre para no confundir archivos.

Aunque estudiaremos la estructura sobre los problemas recogidos como aplicaciones en el capítulo siguiente, podemos afirmar que el archivo de texto del modelo consta de las siguientes partes, claramente diferenciadas:

i) Bloque 1: Encabezamiento

Contiene el nombre del problema (1ª línea) y una breve descripción o notas aclaratorias (un número de líneas variable) que sirven para recordar al usuario aspectos generales o específicos del mismo. Estas líneas del encabezamiento (así como alguna otra a lo largo del archivo, van precedidas de un asterisco '*', lo que significa que no afectan al circuito del modelo por lo que pueden ser eliminadas sin modificar éste,

ii) Bloque 2: Parámetros del problema

En este bloque se recoge el listado de los parámetros del problema con sus valores. Cada parámetro se escribe en una línea independiente del archivo de texto, en la forma siguiente:

.PARAM 'nombre del parámetro' = 'Valor del parámetro'

iii) Bloque 3. Circuito principal

Contiene las sentencias descriptivas de los componentes eléctricos del modelo en red contenidos en el circuito principal. Para problemas definidos por una única ecuación, se especifican en primer lugar las fuentes de corriente correspondientes a los términos de la ecuación (existen tantas fuentes como términos contenga la ecuación más la fuente asociada al término o términos independientes (una sola fuente agrupa todos los términos independientes). Todas las fuentes se conectan en paralelo y cada una se especifica con su nombre, los terminales a los que se conecta su salida (uno de ellos el nodo principal cuyo voltaje es la solución de la variable dependiente) y el valor de la corriente de salida; la sintaxis de escritura de las fuentes es (v. manuales de Pspice):

‘Nombre de la fuente’ ‘nodo + de salida’ nodo – de salida’ VALUE = {‘valor de la corriente controlada de

Salida de la fuente, escrita con la simbología matemática propia del código’}

Los nombres de las fuentes son G_1, G_2, G_3,... (ver apartado nomenclatura de este capítulo). Además, este bloque contiene una resistencia de elevado valor (teóricamente infinito) en paralelo con las fuentes para establecer condiciones de continuidad en corriente continua entre los nodos; se trata de un requisito para testear el circuito, una prueba impuesta por el código Pspice. La sentencia para esta resistencia es

‘R_inf_p’ ‘nodo + de salida’ nodo – de salida’ ‘Valor de la resistencia’

iv) Bloque 4. Circuitos auxiliares

En este bloque se escriben las sentencias asociadas a los dispositivos eléctricos de los circuitos auxiliares que implementan las sucesivas derivadas. Para problemas definidos por una única ecuación diferencial el número de circuitos coincide con el orden de la ecuación. Los circuitos auxiliares se escriben seguidos, empezando con el correspondiente a la primera derivada, a continuación el de la segunda y así sucesivamente. Cada circuito auxiliar consta de una fuente de corriente controlada y sus correspondientes componentes que se especifican mediante sentencias dentro del sub-bloque. La

nomenclatura de las fuentes es E_1, H_1, H_2, H_3... Tras la descripción de los n circuitos auxiliares del problema (con n el orden de la ecuación), se incluye un último circuito que convierte el valor de la segunda derivada de la función en una magnitud tensión eléctrica; este circuito contienen una fuente de corriente controlada, H_n y una resistencia unidad.

El primer circuito auxiliar transforma la variable dependiente de una magnitud tensión (en el circuito principal) a una magnitud tensión (mediante la fuente controlada E_1), que aplica a su correspondiente condensador, C_E1, para general la primera derivada como corriente en este elemento, lee esta corriente (mediante una pila en serie con el condensador, V_nula_E1) y la transforma en tensión en los extremos de la fuente de siguiente circuito auxiliar (H_1) que vuelve a repetir el proceso sucesivamente hasta obtener leer la última derivada como corriente en el último condensador (C_Hn-1). Esta corriente, se lleva a la última fuente (H_n) que la transforma en tensión en los extremos de la resistencia unida (R_uni_Hn) conectada a dicha fuente. La sintaxis de escritura de los componentes es la misma que la explicada para el circuito principal. Los nodos activos de las fuentes auxiliares tienen la sintaxis N_aux_'nombre de la fuente'. Finalmente, como en el circuito principal, cada circuito auxiliar (menos el último) contiene una resistencia infinita en paralelo para asegurar la continuidad.

En problemas formulados por ecuaciones acopladas se sigue la secuencia: circuito principal y circuitos auxiliares de la primera variable (fuentes controladas y componentes), ídem de la segunda variable, y así sucesivamente.

v) Bloque 5. Circuito auxiliar generador de la variable tiempo

Contiene la fuente lineal a tramos generadora de t, conectada al nodo activo N_t, donde se lee el tiempo en

forma de tensión, y una resistencia unidad en paralelo. La sintaxis de la fuente y de la resistencia es:

V_timeN_t 0 PWL(0,0 t_{máx},t_{máx})

R_timeN_t 0 1

dondet_{máx} indica el valor máximo de la ventana de tiempos de la simulación.

vi) Bloque 6. Sentencias auxiliares

Constituyen un conjunto de sentencias necesarias para especificar aspectos complementarios de la simulación. Entre éstas cabe destacar las sentencias '.TRAN' y '.PRINT TRAN' cuyas sintaxis son

.TRAN 'valor del intervalo de tiempo en la impresión de datos de salida' 'tiempo final de simulación'

'tiempo de comienzo de impresión de datos de salida' 'valor máximo del tiempo de paso en el cálculo'

.PRINT TRAN 'nodo cuya tensión o componente cuya corriente se desea imprimir' 'nodo cuya tensión o

componente cuya corriente se desea imprimir' 'nodo cuya tensión o componente cuya corriente s

desea imprimir'...

CODENS_14 incorpora automáticamente el resto de las sentencias necesarias para la simulación.

En el Capítulo de aplicaciones se muestran los archivos '.cir' de los problemas estudiados. Allí se comentarán aspectos complementarios relacionados con la estructura de los archivos de texto de los modelos.

4.4 Bloque de programación (GUI) y archivos (m-file) contenidos en CODENS_13

El programa que se presenta integra los siguientes bloques de programación con un contenido global de 56308 líneas de programación:

i) Para la solución de ecuaciones diferenciales:

CODENS.m/CODENS.fig	757	líneas de programación
normal.m/normal.fig	2743	"
Param.m/Param.fig	1710	"
EjecutaCIR.m	81	"
EjecutaTXT.m	4	"
graficasnormal.m/graficasnormal.fig	364	"

i) Para la solución de ecuaciones diferenciales acopladas:

acoplada.m/acoplada.fig	10456	líneas de programación
acoplada1.m/acoplada1.fig	4593	“
acoplada2.m/acoplada2.fig	4593	“
acoplada3.m/acoplada3.fig	4593	“
acoplada4.m/acoplada4.fig	4593	“
acoplada5.m/acoplada5.fig	4593	“
acoplada6.m/acoplada6.fig	4593	“
acoplada7.m/acoplada7.fig	4593	“
acoplada8.m/acoplada8.fig	4593	“
EjecutaCIRacoplada.m	1677	“
EjecutaTXTacoplada.m	4	“
graficasacop.m/graficasacop.fig	1768	“

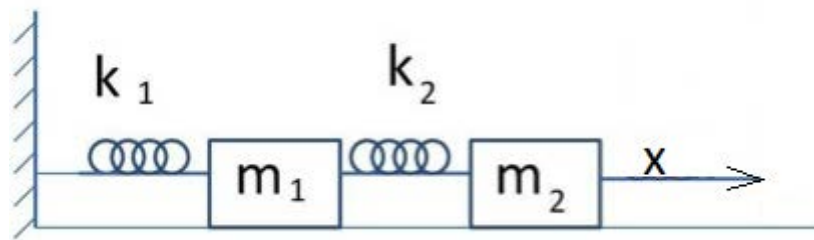
APLICACIÓN DE EJEMPLOS

Capítulo 5

Procederemos a resolver dos problemas. Todos los problemas consisten en ecuaciones acopladas, de índole más compleja que los de ecuaciones ordinarias y por tanto muy adecuados para resolverlos con CODENS. El primero corresponde a masas alineadas unidas entre sí y a una pared mediante muelles, con dos ecuaciones acopladas; el segundo es un problema de modelo biológico de tres especies, con tres ecuaciones acopladas.

MASAS ALINEADAS UNIDAS ENTRE SÍ Y A UNA PARED MEDIANTE MUELLES

Sean dos masas de dimensiones despreciables y valores m_1 y m_2 unidas entre sí mediante un muelle; a su vez, una de ellas se sujeta a una pared fija, también mediante un muelle. Se restringe el movimiento a la dirección normal a la pared.



Los dos grados de libertad se definen mediante las variables x_1 y x_2 , posición de las masas en relación de la posición de las mismas en equilibrio definida por x_a y x_b . Los muelles tienen un comportamiento lineal, $f_m = k(l - l_0)$, con k la constante elástica, l la longitud actual del muelle y l_0 la longitud en reposo. Para simplificar el problema asumiremos que la oscilación del sistema es sólo debida a un desplazamiento inicial de las masas (de una o ambas), sin que éstas estén animadas de velocidad inicial.

Las ecuaciones de gobierno son:

$$m_1 \left(\frac{d^2 x_1}{dt^2} \right) + (k_1 + k_2)x_1 - k_2 x_2 - k_1 A \sin(wt) = 0$$

$$m_2 \left(\frac{d^2 x_2}{dt^2} \right) + k_2 x_2 - k_1 x_1 = 0$$

Los parámetros del problema son $m_1=0.4$; $m_2=0.3$; $k_1=0.6$; $k_2=0.6$, $w=12$, $A=0.01$.

El resto de parámetros son:

```

relatole(RELTOL o tolerancia relative) = 0.0001;
print_step (intervalo de impresión de los datos de salida de simulación) = 1;
results_delay (tiempo inicial para la impresión de los datos anteriores) = 0;
step_ceiling (intervalo máximo del tiempo de paso en el cálculo numérico) =
0.40000;
y1_ini (valor inicial de la variable y) = 0.01;
d1y1_ini (valor inicial de su derivada) = 0;
y2_ini (valor inicial de la variable y) = 0.01;
d1y2_ini (valor inicial de su derivada) = 0;
tmax (tiempo máximo de simulación) = 500 s.

```

El archivo de texto del modelo creado por CODENS_14 es:

```

** MASAS ALINEADAS UNIDAS ENTRE SI Y A UNA PARED MEDIANTE MUELLES **
**

**

**PARAMETROS**

.PARAM m1=0.4
.PARAM m2=0.3
.PARAM k1=0.6
.PARAM k2=0.6
.PARAM A=0.01
.PARAM w=12
.PARAM tmax = 500.000000
.PARAM relatole = 0.0001
.PARAM print_step = 1.000000
.PARAM results_delay = 0.000000
.PARAM step_ceiling = 10.000000
.PARAM y1_ini = 0.01

```

```

.PARAM d1y1_ini = 0

.PARAM y2_ini = 0.01

.PARAM d1y2_ini = 0

**

*CIRCUITO PRINCIPAL 1ª ECUACION*

**

G_2_y1 1N_p_y1 0 VALUE={{(m1)*(V(1N_H2_y1,0))}}
G_1_y1 1N_p_y1 0 VALUE={{0*((V(1N_H1_y1,0))^0)}}
G_0_y1 1N_p_y1 0 VALUE={{(k1+k2)*((V(1N_E1_y1,0))^(1))}}
G_i_y1 1N_p_y1 0 VALUE={{((-k2)*(V(1N_p_y2,0))-k1*A*SIN(w*V(1N_t,0)))}}
R_inf_p_y1 1N_p_y1 0 1E10

**

*CIRCUITO AUXILIARES 1ª ECUACION*

**

E_1_y1 1N_E1_y1 0 VALUE={V(1N_p_y1,0)}
C_E1_y1 1N_E1_y1 1N_aux_E1_y1 1 IC={0.01}
V_NULA_E1_y1 0 1N_aux_E1_y1 0
R_inf_a_E1_y1 1N_E1_y1 0 1E10

**

H_1_y1 0 1N_H1_y1 V_NULA_E1_y1 1
C_E2_y1 1N_H1_y1 1N_aux_H1_y1 1 IC={0}
V_NULA_H1_y1 0 1N_aux_H1_y1 0
R_inf_a_H1_y1 1N_H1_y1 0 1E10

**

H_2_y1 0 1N_H2_y1 V_NULA_H1_y1 1
R_aux_H2_y1 1N_H2_y1 0 1

**

*CIRCUITO PRINCIPAL 2ª ECUACION*

```

**

G_2_y2 1N_p_y2 0 VALUE={{m2}*(V(1N_H2_y2,0))}

G_1_y2 1N_p_y2 0 VALUE={0*((V(1N_H1_y2,0))^0)}

G_0_y2 1N_p_y2 0 VALUE={{k2}*((V(1N_E1_y2,0))^(1))}

G_i_y2 1N_p_y2 0 VALUE={{-k1*(V(1N_p_y1,0))}}

R_inf_p_y2 1N_p_y2 0 1E10

**

CIRCUITO AUXILIARES 2ª ECUACION

**

E_1_y2 1N_E1_y2 0 VALUE={V(1N_p_y2,0)}

C_E1_y2 1N_E1_y2 1N_aux_E1_y2 1 IC={0.01}

V_NULA_E1_y2 0 1N_aux_E1_y2 0

R_inf_a_E1_y2 1N_E1_y2 0 1E10

**

H_1_y2 0 1N_H1_y2 V_NULA_E1_y2 1

C_E2_y2 1N_H1_y2 1N_aux_H1_y2 1 IC={0}

V_NULA_H1_y2 0 1N_aux_H1_y2 0

R_inf_a_H1_y2 1N_H1_y2 0 1E10

**

H_2_y2 0 1N_H2_y2 V_NULA_H1_y2 1

R_aux_H2_y2 1N_H2_y2 0 1

**

V_time 1N_t 0 PWL(0,0 500.000000,500.000000)

R_time 1N_t 0 1

**

.TRAN 1.000000 500.000000 0.000000 10.000000

.PRINT TRAN V(1N_p_y1,0) V(1N_H1_y1,0) V(1N_H2_y1,0)


```
.PRINT TRAN V(1N_p_y2,0) V(1N_H1_y2,0) V(1N_H2_y2,0)

.OPTIONS RELTOL 0.0001

.OPTIONS NUMDGT 5

.PROBE

.END
```

Una vez introducidos los parámetros y las ecuaciones procedemos a realizar la simulación con el programa CODENS_14, obteniendo las siguientes representaciones gráficas en función de los parámetros que deseamos seleccionar:

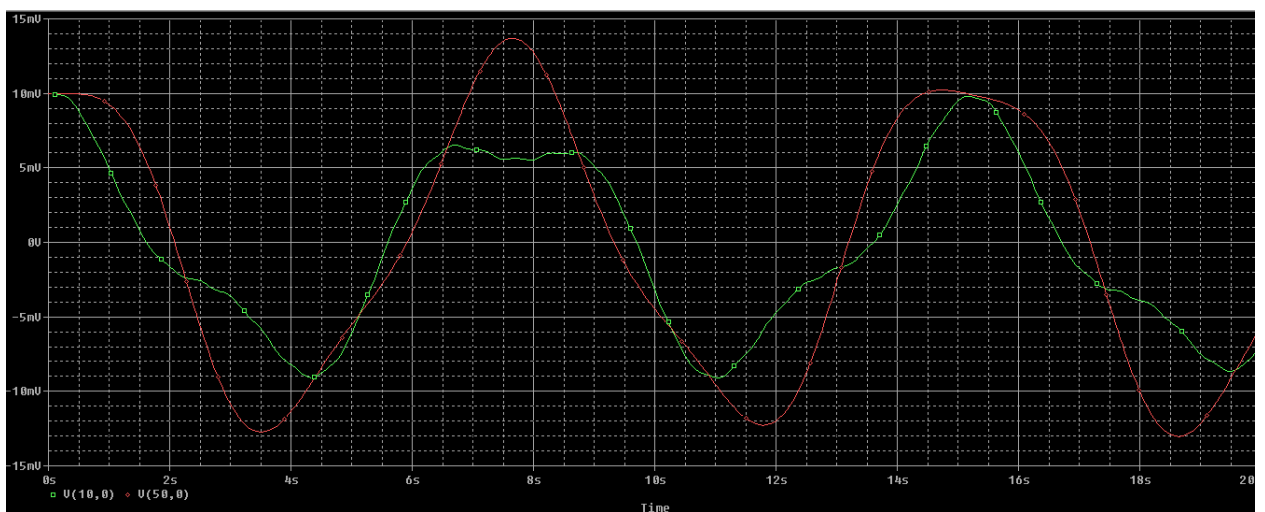


Figura 5.1.1 Solución $x_1(t)$ y $x_2(t)$

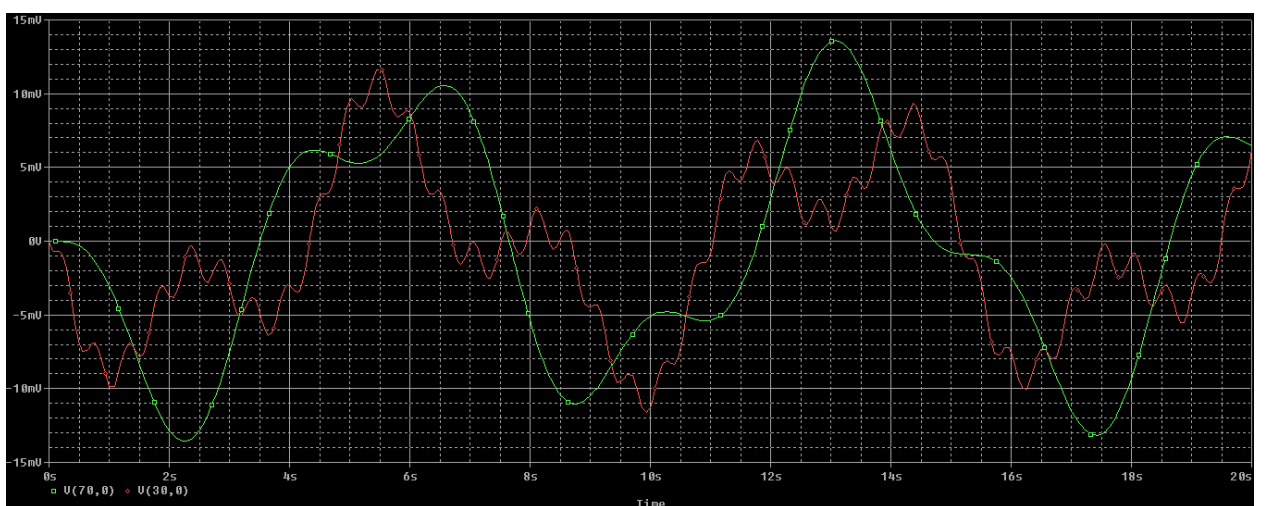


Figura 5.1.2 Solución $x_1'(t)$ y $x_2'(t)$

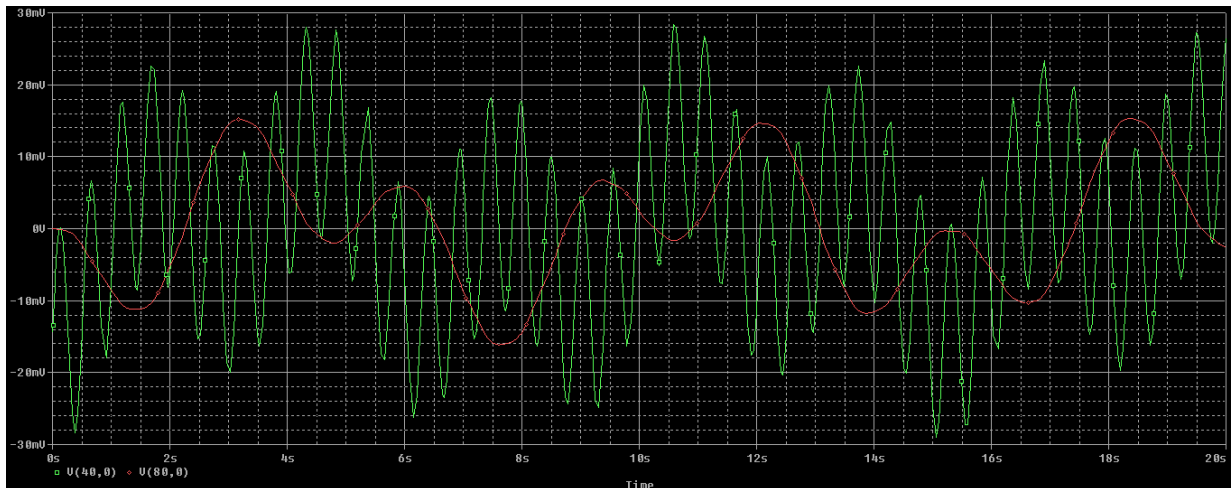


Figura 5.1.3 Solución $x_1''(t)$ y $x_2''(t)$

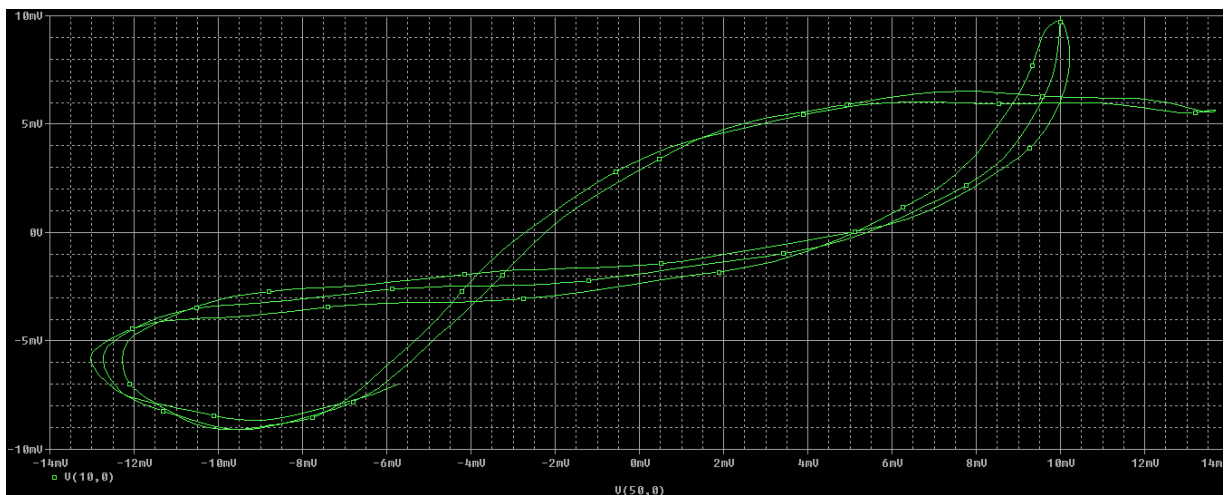


Figura 5.1.4 Solución x_1 frente a x_2

MODELO BIOLÓGICO DE TRES ESPECIES

Se trata de un problema de interacción entre tres especies, regido por ecuaciones acopladas, no lineales, formuladas generalmente en primeras derivadas (problemas de Lotka-Volterra). No incluyen términos de nacimiento y muerte de las especies sino sólo los asociados a la interacción mutua. Las especies más débiles desaparecerán gradualmente mientras que las más fuertes estacionarán su valor con la misma cadencia temporal.

Para las tres especies que interaccionan, las ecuaciones de gobierno serán:

$$\left(\frac{dx}{dt}\right) = -a_0xyz$$

$$\left(\frac{dy}{dt}\right) = b_0xyz$$

$$\left(\frac{dz}{dt}\right) = c_0xyz$$

Los parámetros del problema son $a=-1$; $b=2$; $c=3$.

El resto de parámetros son:

relatole(RELTOL o tolerancia relative) = 0.00001;

print_step (intervalo de impresión de los datos de salida de simulación) = 0.001;

results_delay (tiempo inicial para la impresión de los datos anteriores) = 0;

step_ceiling (intervalo máximo del tiempo de paso en el cálculo numérico) = 0.0010;

y1_ini (valor inicial de la variable y) = 5;

y2_ini (valor inicial de la variable y) = 2;

tmax (tiempo máximo de simulación) = 0.05s.

El archivo de texto del modelo creado por CODENS_14 es:

**** MODELO BIOLÓGICO TRES ESPECIES ****

****PARAMETROS****

.PARAM a=-1

.PARAM b=2

.PARAM c=3

.PARAM tmax = 0.050000

.PARAM relatole = 0.00001

.PARAM print_step = 0.001000

.PARAM results_delay = 0.000000

.PARAM step_ceiling = 0.001000

```

.PARAM y1_ini = 5

.PARAM y2_ini = 2

**

*CIRCUITO PRINCIPAL 1º ECUACION*

**

G_1_y1 1N_p_y1 0 VALUE={{1}}*(V(1N_H1_y1,0))}

G_0_y1 1N_p_y1 0
VALUE={{(a)*(V(1N_p_y2,0))*(V(1N_p_y3,0))}*((V(1N_E1_y1,0))^(1))}

G_i_y1 1N_p_y1 0 VALUE={0}

R_inf_p_y1 1N_p_y1 0 1E10

**

*CIRCUITO AUXILIARES 1º ECUACION*

**

E_1_y1 1N_E1_y1 0 VALUE={V(1N_p_y1,0)}

C_E1_y1 1N_E1_y1 1N_aux_E1_y1 1 IC={5}

V_NULA_E1_y1 0 1N_aux_E1_y1 0

R_inf_a_E1_y1 1N_E1_y1 0 1E10

**

H_1_y1 0 1N_H1_y1 V_NULA_E1_y1 1

R_aux_H1_y1 1N_H1_y1 0 1

**

*CIRCUITO PRINCIPAL 2º ECUACION*

**

G_1_y2 1N_p_y2 0 VALUE={{1}}*(V(1N_H1_y2,0))}

G_0_y2 1N_p_y2 0 VALUE={{(-
b)*(V(1N_p_y1,0))*(V(1N_p_y3,0))}*((V(1N_E1_y2,0))^(1))}

G_i_y2 1N_p_y2 0 VALUE={0}

R_inf_p_y2 1N_p_y2 0 1E10

**

```

CIRCUITO AUXILIARES 2º ECUACION

**

E_1_y2 1N_E1_y2 0 VALUE={V(1N_p_y2,0)}

C_E1_y2 1N_E1_y2 1N_aux_E1_y2 1 IC={2}

V_NULA_E1_y2 0 1N_aux_E1_y2 0

R_inf_a_E1_y2 1N_E1_y2 0 1E10

**

H_1_y2 0 1N_H1_y2 V_NULA_E1_y2 1

R_aux_H1_y2 1N_H1_y2 0 1

**

CIRCUITO PRINCIPAL 3º ECUACION

**

G_1_y3 1N_p_y3 0 VALUE={{1}*V(1N_H1_y3,0)}}

G_0_y3 1N_p_y3 0 VALUE={{(-
c)*V(1N_p_y1,0))*V(1N_p_y2,0))*((V(1N_E1_y3,0))^(1))}}

G_i_y3 1N_p_y3 0 VALUE={0}

R_inf_p_y3 1N_p_y3 0 1E10

**

CIRCUITO AUXILIARES 3º ECUACION

**

E_1_y3 1N_E1_y3 0 VALUE={V(1N_p_y3,0)}

C_E1_y3 1N_E1_y3 1N_aux_E1_y3 1 IC={0}

V_NULA_E1_y3 0 1N_aux_E1_y3 0

R_inf_a_E1_y3 1N_E1_y3 0 1E10

**

H_1_y3 0 1N_H1_y3 V_NULA_E1_y3 1

R_aux_H1_y3 1N_H1_y3 0 1

**

```

V_time 1N_t 0 PWL(0,0 0.050000,0.050000)
R_time 1N_t 0 1
**

.TRAN 0.001000 0.050000 0.000000 0.001000
.PRINT TRAN V(1N_p_y1,0) V(1N_H1_y1,0)
.PRINT TRAN V(1N_p_y2,0) V(1N_H1_y2,0)
.PRINT TRAN V(1N_p_y3,0) V(1N_H1_y3,0)
.OPTIONS RELTOL 0.00001
.OPTIONS NUMDGT 5
.PROBE
.END

```

Con las ecuaciones y los parámetros ya introducidos, procedemos a realizar la simulación con el programa CODENS_14, obteniendo las siguientes representaciones gráficas en función de los parámetros que deseamos seleccionar:

La primera simulación corresponde con el conjunto de valores:

$a=1$; $b=2$; $c=3$

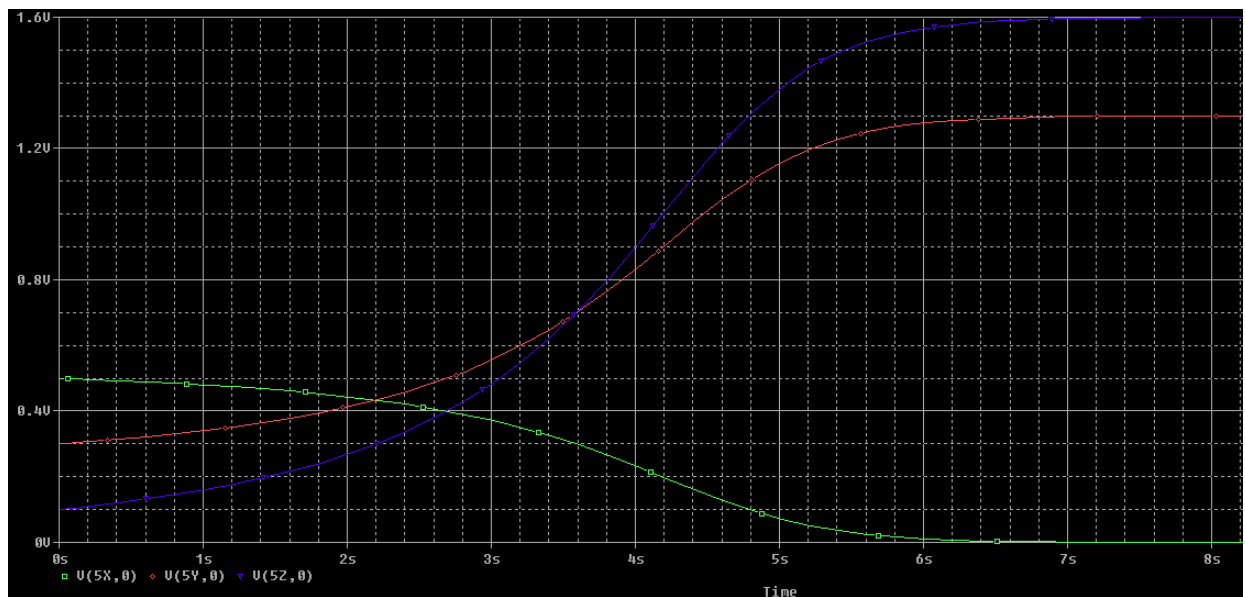


Figura 5.2.1 $x(t)$, $y(t)$ y $z(t)$

La siguiente simulación pertenece al conjunto de valores:

$a=0.5$; $b=1$; $c=1.5$

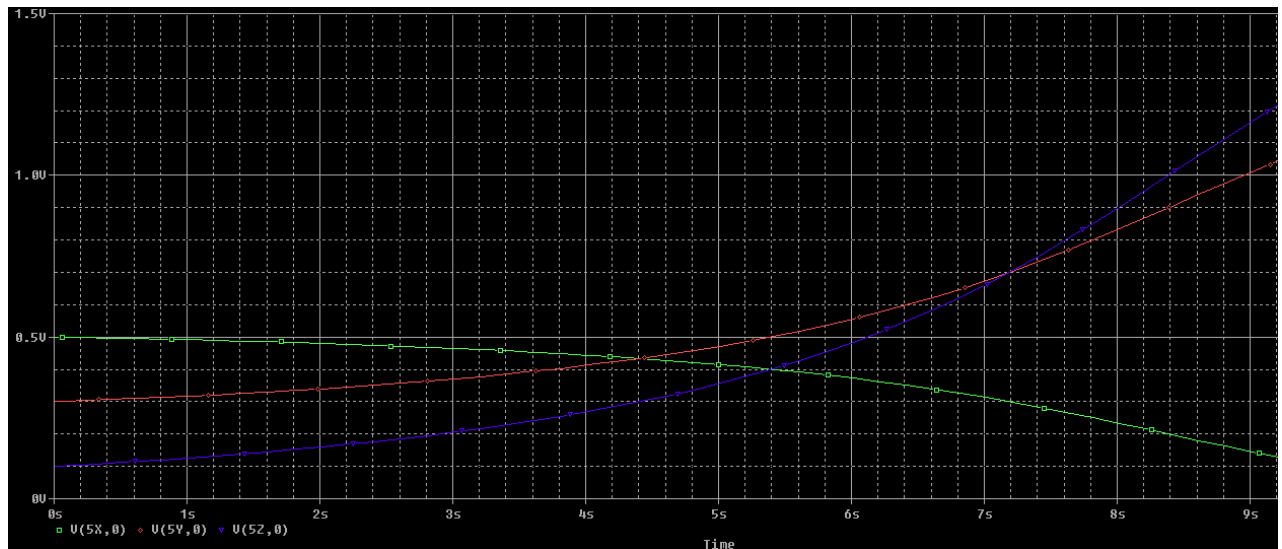


Figura 5.2.3 $x(t)$, $y(t)$ y $z(t)$

Como podemos observar la elección de los parámetros es significativa, particularmente para ver la evolución de las especies al principio del intervalo temporal.

CONCLUSIÓN

Capítulo 6

A través del método de simulación por redes y la analogía eléctrica inherente, hemos podido diseñar modelos fiables de problemas formulados mediante ecuaciones diferenciales ordinarias de cualquier orden, grado y tipo de no linealidad.

De la misma forma se ha podido diseñar modelos para problemas formulados mediante ecuaciones diferenciales ordinarias, acopladas, de hasta orden cuatro, ocho variables, grado arbitrario y cualquier tipo de no linealidad. La extensión a un mayor número de variables y orden de la ecuación no constituye un problema, habiéndose establecido el límite por cubrir con suficiencia el tipo de ecuaciones que surgen en ingeniería y ciencias.

Los modelos anteriores, bajo una nomenclatura intuitiva y adecuada para todos sus dispositivos eléctricos y nodos, y mediante un interface de comunicación para introducción de datos programado en MATLAB, se han convertido en archivos de texto ejecutables en un código de simulación de circuitos (Pspice) y representados, también mediante rutinas adecuadas de procesamiento de datos de Matlab, en el ambiente gráfico de este programa.

Todo los pasos anteriores: i) interface de entrada de datos, ii) elaboración del archivo de texto del modelo, iii) ejecución en Pspice, iv) procesamiento y representación en el entorno Matlab, se han integrado en un nuevo programa, CODENS_14 de más de 55000, elaborado con Matlab.

Por un lado, la fiabilidad del método de redes ha sido contrastada en numerosos trabajos de investigación llevados a cabo por el grupo de simulación de redes de la Upct, por otro, el correcto trabajo de CODENS_14, en cuanto a la elaboración de los modelos, ha sido comprobado con numerosas aplicaciones de las cuales se muestran en este trabajo sólo tres, a título ilustrativo. Así mismo, se ha procedido a mejorar la interface, reducir el número de errores y hacer más intuitivo el programa. Por ello,

podemos decir que CODENS_14 es programa muy interesante, capaz de cubrir con creces los objetivos para los que ha sido elaborado.

Referencias

Alhama, F. Estudio de respuestas térmicas transitorias en procesos no lineales de transmisión de calor mediante el método de simulación por redes. Tesis doctoral, Universidad de Murcia (1999)

Alhama I., Soto Meca A. Rodríguez Estrella T. y Alhama, F. (2010). Characterization of scenarios of salt intrusion in coastal aquifers with salt flats. Proceedings of the Salt Water Intrusion Meeting, SWIM, Portugal.

González-Fernández C.F., F. Heat Transfer and the Network Simulation Method. Horno J. Ed. ResearchSignpost, Kerala (2002)

Alhama, F. y del Cerro Velázquez, F. Simulación y diseño de problemas de conducción térmica con PROCCA-09 (con CD del programa PROCCA-09). Servicio de Publicaciones de la Universidad de Murcia (2010)

Soto Meca A., Alhama F. y González-Fernández C. F. (2007a). An efficient model for solving density driven groundwater flow problems based on the network simulation method. J. Hidrol., 339, 39-53.

PSICE, versión 6.0: Microsim Corporation, 20 Fairbanks, Irvine, California 2718 (1994)